

有助於提高服務品質的前瞻式封包排程機制

摘要

受到封包網路原本忽略時效性特性之影響，對時效性要求極高的多媒體網路服務，如 Voice over IP (VoIP) 以及 Video on Demand (VoD) 在 All-IP 整合的核心網路上提供時，其服務品質低於傳統之電路交換網路。

封包在網路的傳遞過程中受到各種因素之影響，於到達目的地時，可能會造成 long delay time, high jitter 或 packet loss，而在目的地端幾乎已經沒有補救機會，故而如果能在傳遞的過程中，依封包的時效性及重要性做適度的次序調動(rescheduling)而不要依序傳遞(FIFO)，讓過遲的封包提前送出，而將有時間餘裕的封包稍緩送出，如此截長補短，可提高網路效能及整體 QoS 滿意度。

我們在 BBQ (Budget-Base QoS) 的架構下發展一套簡單而有效的方法，在單佇列(Single Preemptive Queue)及多佇列(Multiple FIFO Queue)的 router 架構下，根據封包時效性及重要性賦予合適的 profit function，並參考封包在後續路程上各 router 的負載狀態以便能更精確預估封包是否能及時到達目的地並調整 profit function 參數以調整封包的送出順序，如此能提高排程

的效能。

我們先對單一服務等級的封包排程進行研究，獲得參數調整之技巧，再根據其結果設計多服務等級的封包排程方法，其重點在於如何調配 profit function 給不同的服務等級。

我們藉由 NS-2 模擬模擬器進行實驗，評估本方法的效能，實驗結果顯示我們的方法可以較每個 router 僅根據自身所知的資訊進行排程更可以有效提高網路效能，且能對不同的服務等級做差異化處理以提高整體 QoS 滿意度。最後在多等級服務的實驗環境及評估指標下，網路高負載的情況，本方法與 Simulated Priority Queue 排程演算法比較可以提升 57% 的整體滿意度。

QoS-Aware Packet Scheduling by Looking Ahead Approach

Abstract

Running time sensitive multimedia services such as Voice-over-IP (VoIP) and Video-on-Demand (VoD) on All-IP networks may have lower quality than that on the traditional circuit-switched networks.

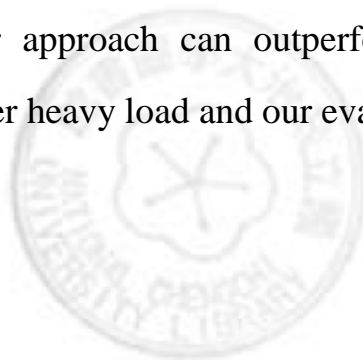
Influenced by many factors, packets transported on a packet-switched network, may suffer from long delay time, large jitter and high packet loss rate. When a packet arrives its destination late, there is no way to correct the problem. Thus, it will be beneficial if routers could forward packets base on their timeliness and importance, instead of using First-In-First-Out (FIFO) service plan, giving important late packets proper precedence. The overall QoS satisfaction will be improved significantly.

In this thesis, we develop a simple and effective scheduling policy based on this concept for the environments where packets have predefined hop-by-hop time schedule. Routers are assumed in two different queue architectures: ideal Single Preemptive Queue router and practical Multiple FIFO Queue router. To forward a packet, a router first assigns a suitable profit function to the packet based on its timeliness and importance as well as the loading status in its succeeding routers along its predefined traveling path, then inserts the packet into an appropriate position in the output queues. Taking the loading status of succeeding routers into account could predict more accurately whether the

packet could reach its destination on time or not.

We conduct the research for the single service class environments first to learn the characteristics of this new scheduling policy, and then for the multiple service class environments based on the knowledge acquired. The challenge is to find the best way to assign proper profit functions to different classes of packets in order to utilize resources more wisely, e.g. urgent and important packets get precedence.

We evaluate the performance of this approach by simulation using NS-2 network simulator. Simulation results show that our approach outperforms our previous version which doesn't take the loading status of succeeding routers into account. Furthermore, our approach can outperform the Simulated Priority Queue by at least 57% under heavy load and our evaluation metrics.



誌謝辭

本論文得以完成，首先要感謝連耀南教授兩年多來的指導，無論是專業知識、邏輯思考、表達能力及論文撰寫等，都讓我獲益良多。還要感謝蔡子傑教授及張宏慶教授平時的指導與協助。

另外，要感謝雷欽隆教授、陳耀宗教授、吳曉光教授及蔡子傑教授能在百忙之中撥冗參加我的論文口試，並指出我未注意到的盲點，使得本論文得以更加完善。

接著要感謝實驗室所有學長、同學及學弟，在這些日子能夠相互砥礪、相互共勉。最後要感謝我的家人朋友及所有幫助過我的人，謝謝！

溫永全 March 14, 2008



目錄

第一章 簡介	1
1.1 封包網路上乘載具時效性應用之問題	1
1.2 UMTS 資料流類別	2
1.3 網路服務品質管理辦法	4
1.4 以預算為基礎之服務品質保證 (Budget Based QoS Management)	5
1.4.1 以 Class 區別的服務策略 (Class Based Service Policies)	5
1.4.2 以預算為基礎之管理 (Budget Based Management)	5
1.4.3 端對端服務品質保證 (End-to-End QoS Management)	6
1.5 Router 架構	6
1.6 排程演算法	7
1.7 研究動機與目的	8
1.8 論文架構	8
第二章 相關研究	9
2.1 A New Jitter Control Mechanism by Per-Hop Packet Scheduling Approach	9
2.1.1 兩種 router 架構	9
2.1.2 獨立排程演算法	11
2.1.2.1 Single Preemptive Queue 獨立排程演算法 (IRS-SPQ)	11
2.1.2.2 Multiple FIFO Queue 獨立排程演算法 (IRS-MQ)	12
2.2 A Priority Based Packet Scheduler with Deadline Considerations	12
2.3 Implementation of Expedited Forwarding Using Dynamic Hop Counts Based Absolute Priority Scheduling	12
2.4 Scalable Hardware Priority Queue Architectures for High-Speed Packet Switches ..	13
2.5 評論	13
第三章 前瞻式排程	15
3.1 Objective Function	16
3.2 排程整體架構	19
3.3 Profit Function	20
3.4 Scheduling Policy	23
3.5 Priority Multiple Queue router (PMQ)	24
3.5.1 PMQ 架構	24
3.6 前瞻式排程 (Looking Ahead Scheduling)	25
3.6.1 設計理念及目標	26
3.6.2 估計後續 router 負載	26
3.6.3 動態調整 profit function 參數	27
3.7 LAS 排程演算法	31

3.7.1 演算法流程	32
3.7.2 演算法虛擬碼 (Pseudo code)	34
第四章 效能評估	35
4.1 實驗環境	35
4.2 實驗設計	36
4.2.1 實驗步驟	37
4.2.2 實驗參數	37
4.3 評估指標	38
4.4 實驗一：在單一服務等級環境中比較 IRS-PMQ 與 IRS-MQ	38
4.4.1 實驗 1A：探討負載對效能的影響	39
4.4.1.1 實驗目標	39
4.4.1.2 實驗流程	39
4.4.1.3 實驗結果與分析	39
4.4.2 實驗 1B：探討路徑長度對效能的影響	41
4.4.2.1 實驗目標	41
4.4.2.2 實驗流程	41
4.4.2.3 實驗結果與分析	42
4.4.3 實驗 1C：探討 queue size 對效能的影響	44
4.4.3.1 實驗目標	44
4.4.3.2 實驗流程	44
4.4.3.3 實驗結果與分析	44
4.5 實驗二：在單一服務等級環境中比較 LAS-PMQ 與 IRS-PMQ	46
4.5.1 實驗 2A：探討負載對效能的影響	47
4.5.1.1 實驗目標	47
4.5.1.2 實驗流程	47
4.5.1.3 實驗結果與分析	47
4.5.2 實驗 2B：探討路徑長度對效能的影響	49
4.5.2.1 實驗目標	49
4.5.2.2 實驗流程	49
4.5.2.3 實驗結果與分析	50
4.5.3 實驗 2C：探討 queue size 對效能的影響	52
4.5.3.1 實驗目標	52
4.5.3.2 實驗流程	52
4.5.3.3 實驗結果與分析	52
4.5.4 實驗 2D：探討他人負載對前瞻式排程效能的影響	54
4.5.4.1 實驗目標	54
4.5.4.2 實驗流程	55

4.5.4.3 實驗結果與分析	56
4.6 實驗三：多服務等級效能評估實驗	57
4.6.1 實驗 3A：探討四種 profit function 共存的效能表現	58
4.6.1.1 實驗目標	58
4.6.1.2 實驗流程	58
4.6.1.3 實驗結果與分析	58
4.6.2 實驗 3B：探討雙 real-time 服務等級共存的效能表現	60
4.6.2.1 實驗目標	60
4.6.2.2 實驗流程	60
4.6.2.3 實驗結果與分析	62
4.7 實驗總結	66
第五章 結論與未來研究方向	67
參考文獻	69



圖目錄

圖 1.1 Router 內部架構	7
圖 1.2 前瞻式排程參考後續 router 負載.....	7
圖 2.1 Single Preemptive Queue router 架構.....	10
圖 2.2 Multiple FIFO Queue router 架構.....	10
圖 3.1 本研究提出的方法	15
圖 3.2 Router 排程之整體架構	19
圖 3.3 Profit function Step-Down.....	21
圖 3.4 Profit function Slope-Down	22
圖 3.5 Profit function Slope-Step	22
圖 3.6 Profit function Double-Slope.....	23
圖 3.7 Single Preemptive Queue 與 Multiple FIFO Queue router 的比較.....	24
圖 3.8 Priority Multiple Queue 中的 Priority Queue	24
圖 3.9 Priority Multiple Queue router 架構	25
圖 3.10 動態調整 profit function 參數的依據.....	27
圖 3.11 封包傳遞中的幾種狀況	28
圖 3.12 調整 profit function 的 local deadline.....	30
圖 3.13 Pre-scheduler 排程演算法流程圖	32
圖 3.14 Post-scheduler 排程演算法流程圖.....	33
圖 4.1 實驗拓樸一	35
圖 4.2 實驗步驟	37
圖 4.3 實驗 1A 變動負載下 IRS-PMQ 與 IRS-MQ 之效能比較	41
圖 4.4 實驗 1B 變動路徑長度下 IRS-PMQ 與 IRS-MQ 之效能比較	43
圖 4.5 實驗 1C 變動 Queue Size 下 IRS-PMQ 與 IRS-MQ 之效能比較.....	46
圖 4.6 實驗 2A 變動負載下 LAS-PMQ 與 IRS-PMQ 之效能比較.....	49
圖 4.7 實驗 2B 變動路徑長度下 LAS-PMQ 與 IRS-PMQ 之效能比較.....	51
圖 4.8 實驗 2C 變動 Queue Size 下 LAS-PMQ 與 IRS-PMQ 之效能比較	54
圖 4.9 實驗拓樸二	55
圖 4.10 實驗 2D 有他人負載下 LAS-PMQ 與 IRS-PMQ 在變動 Queue Size 時的 效能比較	57
圖 4.11 實驗 3A 四種 profit function 共存時變動負載的效能表現	59
圖 4.12 多服務等級實驗拓樸	60
圖 4.13 實驗 3B 雙 real-time 服務等級共存時的效能表現	63
圖 4.14 實驗 3B 不同等級的 throughput	64
圖 4.15 實驗 3B Scheduling policy 1 與 6 的內部結果比較	65

表目錄

表 1.1 UMTS 之 QoS 分類.....	3
表 1.2 各類別之 QoS 品質要求.....	3
表 3.1 Total Charge 的參數.....	17
表 3.2 Quality Index 的參數.....	18
表 3.3 調整 local deadline 的參數及定義.....	29
表 3.4 Local deadline 調整公式.....	30
表 4.1 實驗參數.....	38
表 4.2 實驗 1A 參數.....	39
表 4.3 實驗 1B 參數.....	42
表 4.4 實驗 1C 參數.....	44
表 4.5 實驗 2A 參數.....	47
表 4.6 實驗 2B 參數.....	50
表 4.7 實驗 2C 參數.....	52
表 4.8 實驗 2D 參數.....	55
表 4.9 Scheduling policies for two real-time classes.....	61
表 4.10 評估指標各項參數設定.....	62

第一章

簡介

近年來隨著科技的進步及寬頻網路的普及，通訊網路已經成為日常生活中相當重要的一部份，各式各樣新型的應用服務油然而生，在 All-IP 整合的核心網路上，各種類型的應用皆在同一個封包交換網路上傳送[24]，這些服務對於網路頻寬與服務品質各有不同的要求，舊有網路架構漸漸不敷使用。現今 IP 網路僅提供簡單的 Best Effort 資料流傳送服務，如此的傳送服務並無法針對不同應用類型的封包做差別處理[14]。

網路營運者根據使用者的期望及自身的營運目標，訂定服務品質(QoS)的定義，對於 TSCO (Time Sensitive Connection Oriented)的服務類別而言，其網路服務品質的定義通常是 delay time, jitter 及 packet loss rate 的組合。封包在網路的傳遞過程中受到各種因素之影響，於到達目的地時，可能會造成 long delay time, large jitter 或 high packet loss rate，而在目的地端幾乎已經沒有補救機會，故而如果能在傳遞的過程中，依封包的時效性及重要性做適度的次序調動(rescheduling)而不要依序傳遞(FIFO)，讓過遲的封包提前送出，而將有時間餘裕的封包稍緩送出，如此截長補短，可提高網路效能及整體 QoS 滿意度。

1.1 封包網路上乘載具時效性應用之問題

要在未來的整合型 All-IP 網路上承載所有服務時，受限於封包交換網路的特性，將會面臨以下三大品質問題[1,9]：

A. 封包傳送延遲時間過長 (Long Delay Time)

在 IP 網路中資料是以封包的形式傳送，經過每一個 router 獨處理時間長短不一，導致傳遞延遲時間難以預測。在眾多新型的應用服務當中，有些資料是具有時效性的，資料必須在一定的延遲時間內送達，否則一旦過了時效性，即使最後封包正確送達接收端，對使用者而言也是無效的封包，同時也造成網路資源的浪費。

B. 封包傳送時間抖動劇烈 (Large Jitter)

由於封包的傳遞時間起伏不定，造成一個資料流在傳送封包時，每個封包到達接收端的時間間隔不定，不利於串流服務的傳輸。

C. 封包大量遺失 (High Packet Loss Rate)

封包在傳遞過程中，易因網路阻塞等問題，導致封包在傳送中被丟棄。對於需要可靠資料傳送的應用服務而言，必須重新傳送遺失的封包，如此會導致網路負擔增加。對於不需重傳遺失封包的應用服務而言，其服務品質則將因封包遺失受到損害。

1.2 UMTS 資料流類別

在各式各樣新型的應用服務應運而生之下，各種服務對於 delay time、jitter 以及 packet loss 各有不同的品質要求，因此 3GPP 在 1999 年初所開始制訂的 UMTS 依應用服務類型提出 QoS 分類架構，其中包含對 real-time traffic 的服務考量。UMTS 將目前網路上風行的主要應用依時效性與品質要求概略分為四大類，如表 1.1，分別是交談式 (Conversational class)，串流式(Streaming class)，互動式(Interactive class)，背景式 (Background class)。

表 1.1 UMTS 之 QoS 分類

Class	Example
Conversational	VoIP
Streaming	VoD
Interactive	Telnet
Background	E-Mail

交談式主要用來支援人類雙向溝通，根據人類感官之經驗歸納，此種服務對 delay time 與 jitter 相當敏感，當封包的 delay time 超過 300 ms 時，使用者就難以忍受其通話品質。串流式則要求持續穩定的資料流，因此對 jitter 相當敏感。至於互動式與背景式都屬於 data communication 的服務，可容忍較長的資料傳送時間，但要求所有資料必須正確無誤送達接收端，因此無法忍受資料的遺失。表 1.2 顯示各個類別的品質要求。

表 1.2 各類別之 QoS 品質要求

	Delay Sensitivity	Jitter Sensitivity	Packet Loss Sensitivity
Conversational	High	High	Low
Streaming	Medium	High	Low
Interactive	Medium	Low	High
Background	Low	No	High

1.3 網路服務品質管理辦法

針對以上在封包網路上容易發生的傳送品質問題，網路標準機構 IETF 制定了 Intserv (Integrated service)[6]與 DiffServ (Differentiated service)[4]兩種機制，以提供服務品質保證的環境。

A. Intserv (Integrated Service)

Intserv 使用 RSVP (Resource Reservation Protocol)針對各個訊務建立一保留頻寬的 virtual circuit 來滿足各訊務 QoS 的要求。這樣的架構有其相當的好處，首先，它對使用者提供了絕對的 end-to-end QoS 保證，再者，每個使用者的資料流都可以輕易的監控管理，並且可以運用現有的 routing protocols。但因 overhead 過多，使得網路在擴充性 (scalability)上受到極大的限制。

B. DiffServ (Differentiated Service)

DiffServ 則是採用與 IntServ 不同的策略來提供服務品質保證。DiffServ 是將具有相似 QoS 要求的訊務合併一起處理，對同一類型的資料提供一致性的服務與相對的保證，而非針對個別的訊務提供保證。每一類型的資料會有其相對應的 Per-Hop Behavior 在 DiffServ domain 上傳送。這樣的方法雖然無法達到如 IntServ 般絕對的服務品質保證，但卻可以解決 IntServ 在擴充性和實作上的問題，因此 DiffServ 的架構漸漸取得其主流地位，但如何在 DiffServ 的架構上，提供各個訊務的 end-to-end QoS 保證正是極待解決的主要問題。

Budget-Based QoS Management (BBQ)

在 Yao-Nan Lien、Hung-Ching Jang、Tsu-Chieh Tsai 與 Hsing Luh 發表的論文「Budget Based QoS Management Infrastructure for All-IP Networks」[12]，提出了一套整體的管理機制，名為 Budget Based QoS Management Infrastructure，使得 All-IP 網路可以實現提供

服務保證。我們的研究基於 BBQ 的架構而設計，因此在 1.4 簡要介紹 BBQ 的相關架構。

1.4 以預算為基礎之服務品質保證 (Budget Based QoS Management)

1.4.1 以 Class 區別的服務策略 (Class Based Service Policies)

對於 time-sensitive 和 connection-oriented (TSCO) 的服務，例如 Conversational class 和 Streaming class 的服務需求，BBQ 中的 admission control 會以一個快速的 call setup procedure 來處理，針對服務等級給予服務品質需求的路徑並保留需求的資源，對於其他種類的服務，BBQ 則不保留任何資源，採用 Best-Effort 的政策提供給 time-insensitive 的服務。依據使用者要求的服務品質和願意支付的費用，對應到不同的服務等級，網路營運者可以依此決定服務等級的定價策略，並調校其網管政策來最佳化營運目標。

1.4.2 以預算為基礎之管理 (Budget Based Management)

為了提供端對端網路服務，網路必須提供端對端服務品質保證。在實際的通訊中，一個封包可能必須穿越數個不同營運者的網路，並非侷限於同一網路當中。BBQ 將一個封包所經過的路程，稱為一條 end-to-end path。一條 end-to-end path 包含了許多網路元件，例如接取網路和核心網路。

以預算為基礎之概念就是，品質管理者可以依據網路元件的能力，將使用者所要求的頻寬及服務品質以預算方式分配在這些網路元件上面，而由各網路元件各自負責提供分段的品質保證。

雖然以預算為基礎之管理架構無法達到最佳的整體資源運用效率，但是卻能大幅降低管理複雜度。所以，BBQ 架構使用預算的方式，將管理權責以最佳方式分配到各個網路元件。

1.4.3 端對端服務品質保證 (End-to-End QoS Management)

End-to-end QoS 管理最重要的工作，就是找到符合品質要求的 end-to-end path，BBQ 利用規劃具服務品質保證的路徑，以達到服務品質保證。BBQ 採用透過分層分權的方式，各層元件各自規劃該層級的資源成一路徑片段，提供給上層網路元件規劃成較長之路徑片段。Short-path 為一穿越某一個 Core Network 且提供服務品質保證之路徑；long-path 為一穿越 Backbone 且提供服務品質保證之路徑；end-to-end path 則為 end user 到 end user 且實際提供端對端服務品質保證之路徑。路徑的選擇與資源的配置是採取事先規劃的方式，為每一段時間內可能進入的訊務事先規劃路徑並將資源分配於 Edge Router。

在某些 QoS 管理架構中(例如：IntServ 及 BBQ)，TSCO (Time Sensitive Connection Oriented)類型的封包均循有資源管理的指定路徑傳遞以獲得服務品質之保證，因此可預估這些封包到達每一個 router 的時間及其送出時間。在此情況下，每一個 router 將可根據封包的時刻表、其準時與否的狀態，及其重要性(服務等級)進行次序調動，以期增進整體 QoS 滿意度。

BBQ 計畫的目標是在研究適當的排程機制提供給使用者運用以提高系統效能。在資源不變的情況下，提高 QoS 滿意度，即傳遞封包時，在可接受的 delay time 之內，期望改善封包傳遞至目的地的 jitter 及 packet loss rate。

1.5 Router 架構

圖 1.1 為一般 router 的內部的簡單架構。當封包進入 router 時，router 裡的 packet forwarding 元件會依照封包所要傳送的目的地，選取適當的 output queue 將封包插入，再將 queue 內的封包依序送出，我們的排程方法將對個別的 output queue 中的封包進行排程，決定適當的封包傳遞順序，以求提高網路效能及整體 QoS 滿意度。我們假設兩種不同 output queue 處理方式的 router 架構：single preemptive queue 及 multiple FIFO queue，而我們的

後續研究重點會放在 multiple FIFO queue 的 router 架構之上。

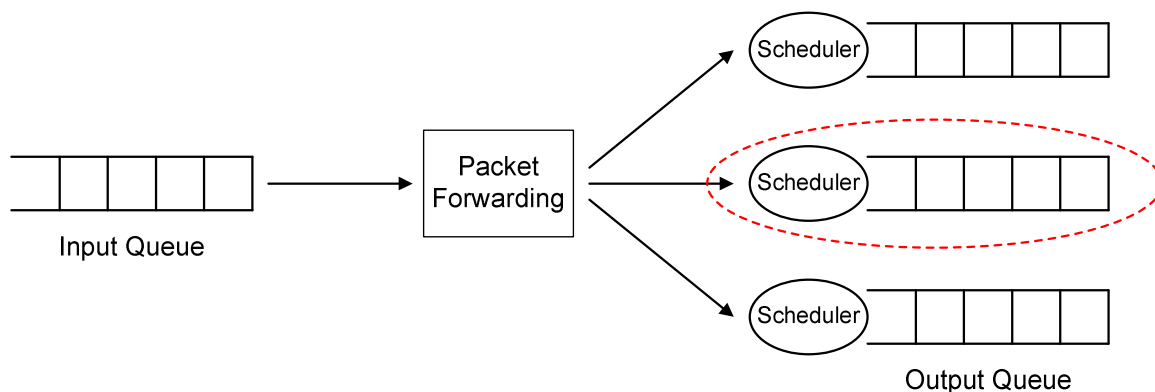


圖 1.1 Router 內部架構

1.6 排程演算法

我們的研究假設兩種排程演算法，一是獨立排程(Intra-Router Scheduling, IRS)，每個 router 根據自身所知的資訊而不參考其他 router 的狀態逕自進行排程，二是前瞻式排程(Looking Ahead Scheduling, LAS)，在此演算法下，router 會參考封包在後續路程上各 router 的狀態進行排程，如圖 1.2 所示。例如一個嚴重遲到的封包如果在後續路程上所經過的 router 上的負載都很重，而判斷無法在限定時間內到達目的地的話，就會在中途被捨棄，反之，如果後續的 router 如果負載很輕的話，則可能會保留該遲到的封包，甚至盡力提前送出，並在後續的 router 逐漸彌補所耽誤掉的時間，以求及時到達目的地。

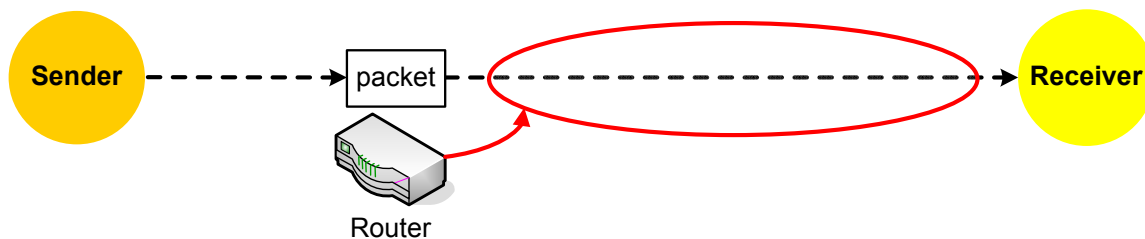


圖 1.2 前瞻式排程參考後續 router 負載

1.7 研究動機與目的

Per-hop QoS 排程是在每一個 hop 內各自進行，如要達到最佳排程，各個 router 之間必須互相分享資訊，並進行分散式運算，此種計算環境在實務上極難實現，此因 router 為了加快處理速度，不可能進行太複雜的運算。

本研究延續 A New Jitter Control Mechanism by Per-Hop Packet Scheduling Approach [13]，該論文提出之排程演算法屬獨立排程(IRS)，我們希望以該論文為研究基礎，並參考封包在後續路程上各 router 的狀態，進而改善其排程演算法，以期前瞻式排程(LAS)可以更有效的提高網路效能及整體 QoS 滿意度。另外針對該論文中的 Multiple FIFO Queue router 架構，我們也提出有效的改善方法。

[13]中所提出的方法屬獨立排程，雖能有效的改善 QoS 滿意度，但有鑒於其未納入考慮的因素，如封包傳遞剩餘路徑上 router 的狀況，我們希望能將這些也一併納入設計演算法的考量。參考封包傳遞剩餘路程上 router 的狀況，可以更有效的了解後續路程的網路狀況、負載，因此我們期望前瞻式排程可以有更好的 QoS、更高的網路使用率。

另外針對[13]中的 Multiple FIFO Queue router 架構，若與 Single Preemptive Queue 架構比較，缺少最前面位置以供插入，導致非常緊急的封包也必須排隊，如此會使效能降低，所以我們提出一個 Priority Multiple Queue router 架構讓其中一個 queue 的優先權提高以供此類封包插入，期望能使其逼近 Single Preemptive Queue 架構之效能。

1.8 論文架構

本文共分為五章，第二章介紹一些近年來對封包排程的相關研究，並分析其優缺點。第三章介紹我們提出的兩個方法。第四章則藉由 NS-2 模擬以驗證我們的方法改善整體 QoS 滿意度的效果。第五章為結論與未來發展。

第二章

相關研究

現今的網路僅提供 Best Effort 的封包傳送服務，router 皆是採取 FIFO 的服務方式，封包到達順序決定其送出順序，並未根據實際需求調整送出的順序，如此的服務方式無法提供良好的品質，因此 QoS 近年來已是被廣泛研究的主题。本研究延續[13]，因此我們將先介紹該篇論文，爾後再介紹其他相關研究，並加以分析其優缺點。

2.1 A New Jitter Control Mechanism by Per-Hop Packet Scheduling Approach

[13]在 BBQ (Budget-Base QoS)的架構下發展一套簡單而有效的方法。假設每個進入 router 的封包都帶有預定送出時間及服務等級，並設計與封包傳遞時間及服務等級相關的四種 profit function，router 將視情況賦予該封包一個 profit function，scheduler 根據封包的 profit function，以最大化總獲利和為目標決定其送出的順序，期望能在最終的接收端呈現較高的服務品質。

該研究假設兩種不同 output queue 處理方式的 router 架構以支援所提出的 per-hop jitter control 機制。

2.1.1 兩種 router 架構

A. Single Preemptive Queue router (SPQ)

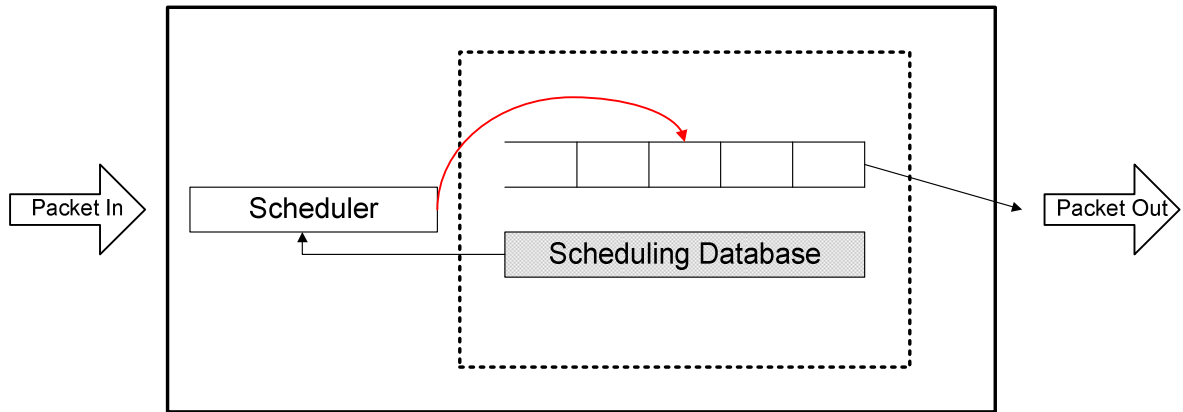


圖 2.1 Single Preemptive Queue router 架構

在此架構下 router 內的個別 output queue 是由 Single Preemptive Queue 所構成，如圖 2.1，而 router 可將進入的封包插到 output queue 的任意位置。但實際網路中的 router 為了加快交換速度不會輕易採取 preemptive queue 作法，且 preemptive queue 在硬體實作上花費也會過高，因此本架構僅能作為參考，實際上被採用之機會不大。

B. Multiple FIFO Queue router (MQ)

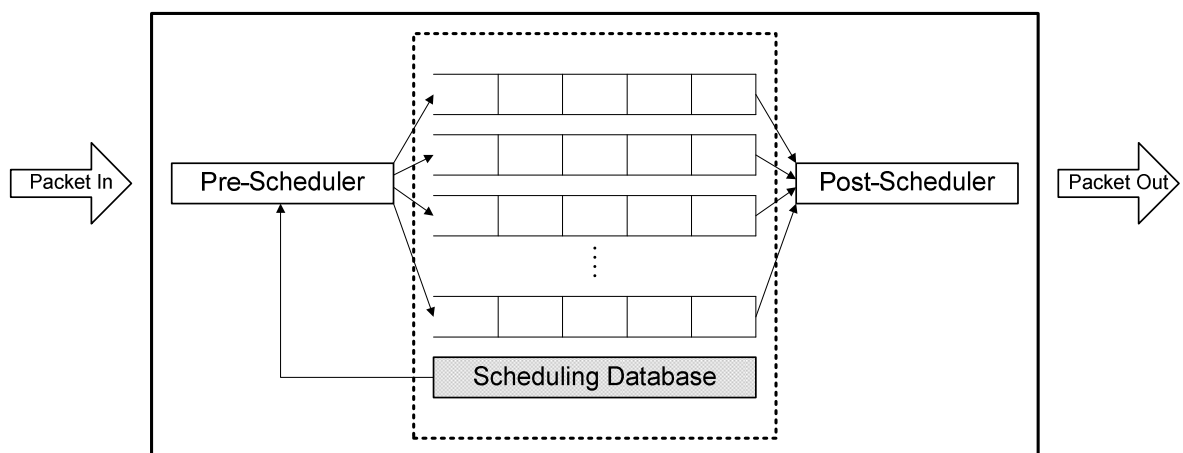


圖 2.2 Multiple FIFO Queue router 架構

在此架構下 router 內的個別 output queue 是由數個 FIFO queue 所組成，如圖 2.2 所示，而 router 只能將進入的封包插到其中一個的尾端。在此架構下尚須設計前端排程器 (pre-scheduler) 及後端排程器 (post-scheduler)，將進入的封包選取適當的位置插入。此種架構較不會影響 router 的處理速度，在實際網路中比較容易被採用，亦是我們後續的研究重點。

2.1.2 獨立排程演算法

對於進入 router 的封包，scheduler 將依據排程演算法插入 queue 中適當的位置，該研究依封包之時效性及服務等級賦予適當的 profit function，並在排程時選擇以最大總獲利為目標的排程，當 router 認為 real-time traffic 的封包無法準時到達接收端時 (profit 為負值)，便將其丟棄。該研究依照以上原則為兩種 router 架構設計排程演算法。

2.1.2.1 Single Preemptive Queue 獨立排程演算法 (IRS-SPQ)

在 Single Preemptive Queue 架構之下，封包可以插入 queue 的任意位置，且每一個封包被賦予一個與傳遞時間及服務等級相關的 profit function，scheduler 依封包 profit function 選擇適當的位置插入使封包送出順序獲得最大總獲利。

[13] 中說明 preemptive 封包排程是一個 NP-hard 的問題，因此必須設計 heuristic solution 以在較短的時間內決定封包送出順序。假設 output queue 大小為 N 個封包，該封包排程演算法將 output queue 內原本 $N-1$ 個封包順序維持不變，但進入 output queue 的封包可以任意插入 queue 之中，最後在此 N 個插入位置中，選取適當的位置插入，使封包送出順序能獲得最大總獲利。此演算法的複雜度為 $O(N)$ 。

2.1.2.2 Multiple FIFO Queue 獨立排程演算法 (IRS-MQ)

在 Multiple FIFO router 架構之下，假設 router 中有數個 FIFO queue，可視為 Single Preemptive Queue Router 的特例，因為只有固定位置(每個 FIFO queue 的尾端)可以插隊。

該排程演算法之 pre-scheduler 依封包之 profit function 選擇適當的 queue 之尾端插入，使得新封包送出順序之總獲利在所有選擇中為最大，而 post-scheduler 部分則將採取輪詢，而將 queue 裡面的封包依序全部送出稱為對該 queue 服務一次。

2.2 A Priority Based Packet Scheduler with Deadline Considerations

[7]中提出兩種演算法 SPD (Static Priority with Deadline Considerations) 與 DPD(Dynamic Priority with Deadline Considerations)，SPD 依據封包的優先權將 queue 中的封包排序，並且會檢查封包的 remaining time to the deadline，若超過 deadline 則會將該封包捨棄，該研究中的 deadline 所指的是 end-to-end 的 deadline，與本研究中到達每個 router 的 local deadline 不同。而 DPD 還會以封包的 remaining time to the deadline 當作修改優先權的依據，再進行排序。並針對降低複雜度提出部分排序的作法。

2.3 Implementation of Expedited Forwarding Using Dynamic Hop Counts Based Absolute Priority Scheduling

[10]中先說明一般 FIFO 的排程演算法下，封包的延遲時間會隨著 hop count 增加而變長；並且不同 hop count 的資料流之間會互相影響，如最大 hop count 的資料流會嚴重影響其他資料流的封包延遲時間，所以提出一個以 hop count 為基準的排程演算法。

該演算法延續他篇研究，主要作法就是提高有較大 hop count 資料流的優先權，並且隨時動態依據剩餘 hop count 調整其優先權，期望能平衡不同 hop count 資料流的封包延遲時間。

2.4 Scalable Hardware Priority Queue Architectures for High-Speed

Packet Switches

[15]中比較不同硬體所架構而成的 Priority Queue 並提出新的硬體架構。Priority Queue 是由多個 queue 組成，每個 queue 有不同的優先順序。相同優先權的 queue 服務同一等級的資料流，低優先權的資料流必須等待高優先權的服務完才能被服務。所以 Priority Queue 可以提供不同服務等級差異化處理。

2.5 評論

[13]提出與封包傳遞時間及服務等級相關的四種 profit function，其中包含數個參數，較為重要的是 hard deadline 及 soft deadline，這裡所提的 deadline 是指封包在個別 router 裡的 hard deadline 及 soft deadline，而非封包到達接收端的 deadline。Step-Down 與 Slope-Step 中超過 hard deadline 的封包將無利可圖，而 Slope-Down 與 Double-Slope 中超過 soft deadline 的封包則是獲利逐漸下降，若超過 hard deadline 的封包同樣無利可圖。

而其參數的設定是採用較為簡單的方式。如 Step-Down 與 Slope-Step 是將 hard deadline 設為其預定送出時間，採取此種設定的時機是當後續的網路狀況非常擁塞而對於遲到的封包沒有補救機會時，如果還有補救機會時，可把 hard deadline 適度延長以增加封包之存活機率；而 Slope-Down 與 Double-Slope 將 soft deadline 設為其預定送出時間，並將 hard deadline 適度延長一段時間，採取此種設定的時機是當後續的網路狀況對於遲到的封包仍有補救機會時。故本研究根據封包在 router 的預定送出時間及後續網路狀況，用以動態調整 profit function 的參數。

觀察該論文所提出的 Step-Down 與 Slope-Step，其設計理念為傳遞中的封包只要過了 hard deadline，則沒有 profit，雖然這樣可以保證接收端不會收到過時的封包，但是做法有些許極端，因為我們可以透過後面的 router 來做補救，所以為了給這類封包補救機

會，本研究依剩餘路徑上 router 的狀況及網路負載做調整，把 hard /soft deadline 適度的延後，使後面的 router 可以有補救機會。雖然該論文也有提出 Slope-Down 與 Double-Slope，乍看之下也有補救的效果，但是並沒有將剩餘路徑上 router 的狀況及網路負載納入考慮。

A Priority Based Packet Scheduler with Deadline Considerations [7]與我們的研究皆是以封包的 deadline 作為封包排程的依據，但該研究並沒有針對不同服務等級的服務提供不同優先次序的服務，一個 All-IP 網路將要承載各種不同種類的服務各自有不同的品質要求，而重要性各有不同。在 router 對封包重新排程時，必須加以考慮不同類型封包的服務等級，給予差異化待遇，讓資源依優先等級調配，如此才可以提高整體滿意度。

Implementation of Expedited Forwarding Using Dynamic Hop Counts Based Absolute Priority Scheduling [10]考慮封包後續 hop count 以進行排程，但 hop count 並不足以反應後續網路狀況，例如剩餘相同 hop count 的後續路程可能壅塞程度不同，無法相提並論。

Priority Queue [15]與本研究相同的是皆使用 multiple FIFO queue router 架構，且都有針對不同服務等級提供不同優先次序服務之概念，但 Priority Queue 中每種等級的 queue 僅服務一種 class，資源無法互相分享，極可能因為某種等級的服務過多影響其他的服務等級，如此會造成整體網路效能低落；而我們提出的方法可讓不同服務等級間可以在統一的調度下共享資源，能有較好的資源使用率。此外，即使是同一種 class 也會因為早到晚到而有所差異，我們的方法也可以做適當的處理。

第三章

前瞻式排程

[13]所提出 per-hop 封包排程的方法是利用 profit function 的方式表達進入一個 router 的封包的緊急程度及服務等級，再配合排程演算法來決定封包排程順序。當封包到達 router 時，router 會依據封包的緊急程度、服務等級以及網路處理能力而賦予封包適當的 profit function，scheduler 再依據封包的 profit function 進行排程，以最大化總獲利和為目標，調整封包送出的順序，期望減小接收端之 jitter，以提高服務品質，該研究著重於 profit function 及排程演算法的設計。我們的研究將延續使用[13]所設計的 profit function。

我們延續[13]並參考封包在後續路程上各 router 的狀態，提出前瞻式排程(Looking Ahead Scheduling)；另外針對該研究中的 Multiple FIFO Queue router 架構提出改善的方法：Priority Multiple Queue router 架構。因此，本研究提出的方法有兩個部份，如圖 3.1 所示，第一個部份為 Priority Multiple Queue router 架構(PMQ)，第二個部份為前瞻式排程(LAS)。本研究所提出的前瞻式排程相對於[13]所提出的獨立排程，最主要的差別是前瞻式排程中 router 會參考封包在後續路程上各 router 的狀態讓 scheduler 能更準確的預估封包即時到達目的地的機會。

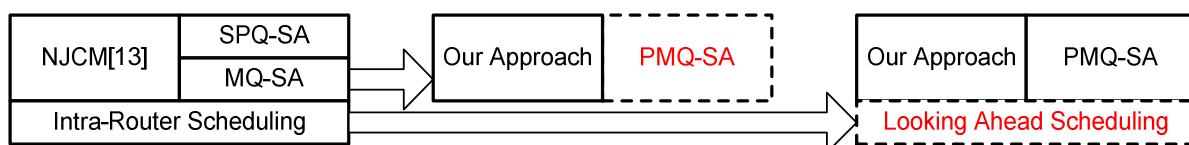


圖 3.1 本研究提出的方法

3.1 Objective Function

服務品質通常是封包延遲時間、jitter 及封包遺失率的組合，我們的研究重點是在於多等級服務(multiple class)的狀況之下找到最佳的 profit function 配方(scheduling policy)，使得整體的滿意度最大。在多等級服務的狀況之下，不同等級服務各有不同的要求，所以我們必須找到一個指標評估整體服務品質。

我們定義了一個統一的量化指標如公式(1)所示，以 total charge(類似 utility function)來做為整體服務品質的量化指標，以便在多服務等級的情況下，衡量使用不同的 profit function 配方組成的不同 scheduling policy 所得到的效能。量化指標的定義當然會影響效能的高低，而不同的綜合指標組合也會產生不同的結果，我們的實驗中只是一個簡單的例子，網路營運者應該視實際情況而有所調整。

在多服務等級的情況下，使用不同的 scheduling policy 將訊務流從起點傳送到目的地所得到的 total charge 如公式(1)所示。假設訊務流分類成 i 種 class，每種 class 各有 j 條訊務流。該公式的基本概念便是分別將 quality index、封包單位價格及封包送出(接收)量相乘，最後加總的結果便是使用該 scheduling policy 的 total charge，我們的目標就是要找出使公式(1)最大的 scheduling policy。其中符號及定義如表 3.1 所示。

表 3.1 Total Charge 的參數

Notation	Definition
$q_{i,j}$	quality of the j-th flow of class i ($0 \leq q \leq 1$)
C_i	unit price per byte of class i
$\lambda_{i,j}$	number of bytes admitted in the j-th flow of class i (Conversational or Streaming class)
	number of bytes received in the j-th flow of class i (Background class)
N_i	number of flows of class i

$$c = \sum_{i=1}^3 \sum_{j=1}^{N_j} (q_{i,j} \cdot C_i \cdot \lambda_{i,j}) \quad (1)$$

公式(1)中針對 class i 中單一訊務流的 quality index 定義如公式(2)所示。其中符號及定義如表 3.2 所示。其中除 α 、 β 、 γ 之外的參數由網路營運者自行依照需求設定。

Quality 之各項參數都有一個最低標準，charge threshold，參數值如低於對應的標準則視為不可接受，滿意度定為 0；各參數均符合 charge threshold 的訊務流則根據其服務等級各有相對應的品質滿意度定義如公式(2)所示，其中 $B_i^{(1)}$ 、 $B_i^{(2)}$ 及 $B_i^{(3)}$ 就是 charge threshold。

表 3.2 Quality Index 的參數

Notation	Definition
$\alpha_i = 1 -$ (average delay time of class i) / $B_i^{(1)}$	delay time satisfaction factor of class i
$\beta_i = 1 -$ (average jitter of class i) / $B_i^{(2)}$	jitter satisfaction factor of class i
$\gamma_i = 1 -$ (packet loss rate of class i)	loss rate satisfaction factor of class i
a_i	quality coefficient of delay time for class i
b_i	quality coefficient of jitter for class i
c_i	quality coefficient of loss rate for class i
$B_i^{(1)}$	maximum acceptable average delay time of class i
$B_i^{(2)}$	maximum acceptable average jitter of class i
$B_i^{(3)}$	maximum acceptable loss rate of class i

$$q_i = \begin{cases} a_i \cdot \alpha_i + b_i \cdot \beta_i + c_i \cdot \gamma_i & \text{for average delay time of class i} \geq B_i^{(1)} \\ & \text{or average jitter of class i} \geq B_i^{(2)} \\ & \text{or loss rate of class i} \geq B_i^{(3)} \end{cases} \quad (2)$$

3.2 排程整體架構

圖 3.2 是我們方法中 router 排程的整體架構。其中 output queue 可以為[13]中的 Single Preemptive Queue(SPQ)、Multiple FIFO Queue(MQ)或本研究的 Priority Multiple Queue(PMQ)。而排程演算法可以為[13]中的獨立排程(IRS)或本研究的前瞻式排程(LAS)。

其中針對封包傳遞時間及服務等級所設計的四種 profit function，在一般多服務等級的網路環境中，我們便從這些 profit function 中選擇出不同的組合分別賦予不同 real-time 服務等級的封包，不同的組合即構成一種 scheduling policy。最終目的即是要決定能夠使整體網路效能最大的封包送出順序。

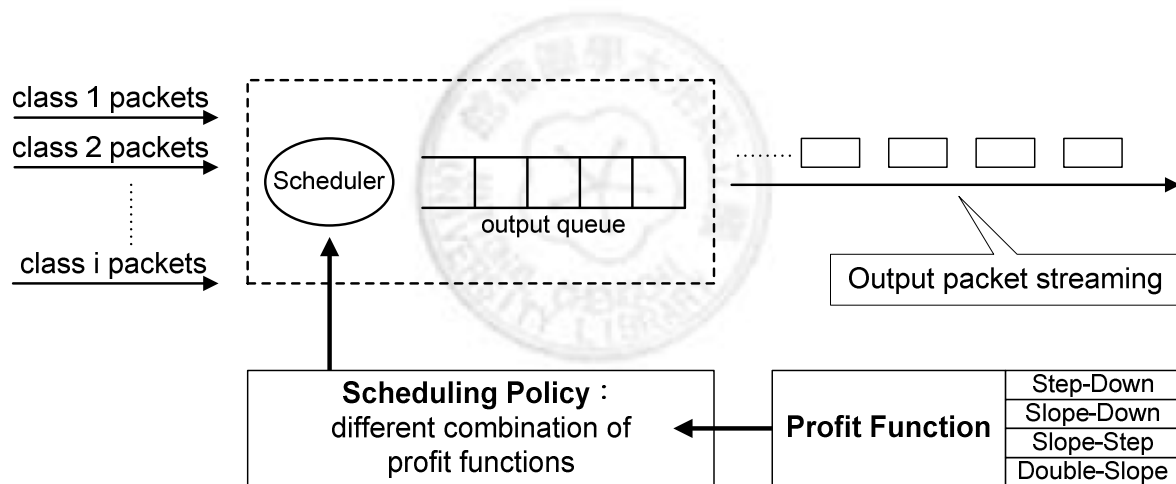


圖 3.2 Router 排程之整體架構

我們將先介紹[13]所提出的四種 profit function 及由不同 profit function 所組成的 scheduling policy。再介紹本研究所提出的 Priority Multiple Queue router 架構(PMQ)及前瞻式排程(LAS)。最後介紹 LAS 排程演算法流程及虛擬碼。

3.3 Profit Function

論文[13]所設計的 profit function 力求簡單有效，避免造成 router 太大負擔，有四個參數如下所述。其中 hard deadline 及 soft deadline 是指封包在個別 router 裡的 hard deadline 及 soft deadline，而非封包到達接收端的 deadline。

A. Hard deadline

超過 hard deadline 的封包將無利可圖，必須被捨棄。本研究假設每一個 real-time 封包都有一個到達接收端的 deadline；hard local deadline 之設定是根據遲到或早到之情況，如果封包無法在 hard local deadline 之前送出，則 router 認為該封包無法如期送達接收端，便將其丟棄，不必再浪費網路資源。

如何設定 hard local deadline 對判斷排程方法之效能有極大影響。

B. Soft deadline

超過 soft deadline 的封包獲利將會下降。如果封包無法在 soft local deadline 之前送出，router 認為封包可能面臨遲到的風險，雖然不會丟棄該封包，但獲利會逐漸降低。

如同 hard deadline，如何設定 soft local deadline 對判斷排程方法之效能有極大影響。

C. Pre soft deadline profit rate

在 soft deadline 之前送出封包時，可獲得之 profit。

D. Post soft deadline profit rate

在 soft deadline 之後送出封包時，可獲得之 profit。

圖 3.3 到圖 3.6 列出四種 profit function 的說明及其示意圖。其中 T_c 為目前時間，而 T_d 為封包預定從該 router 送出之時間，每個封包在傳遞路徑上的每一個 router 到達與送出時間已預定，而由於 IP 封包的傳遞時間起伏不定，會使得封包已傳遞時間不一定等於預定送出時間。[13]中參數的設定是採取簡單的方法，即是將 hard/soft deadline 設為

封包預定在該 router 之送出時間。

A. Step-Down :

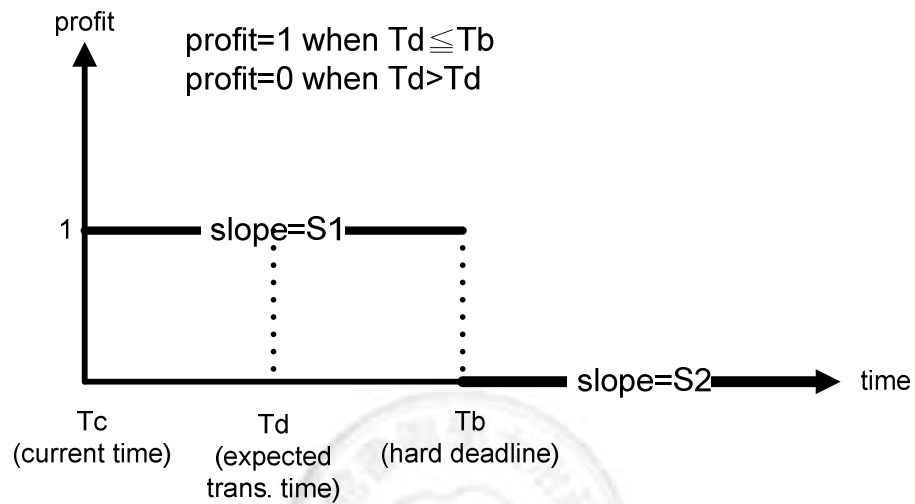


圖 3.3 Profit function Step-Down

第一種 profit function 是 Step-Down，當 router 採用 Step-Down type 的 profit function 處理封包時，scheduler 只須在 hard deadline 之前送出封包即可，提早送出並無額外利益，若無法在 hard deadline 之前送出則無 profit，便將之丟棄。

B. Slope-Down :

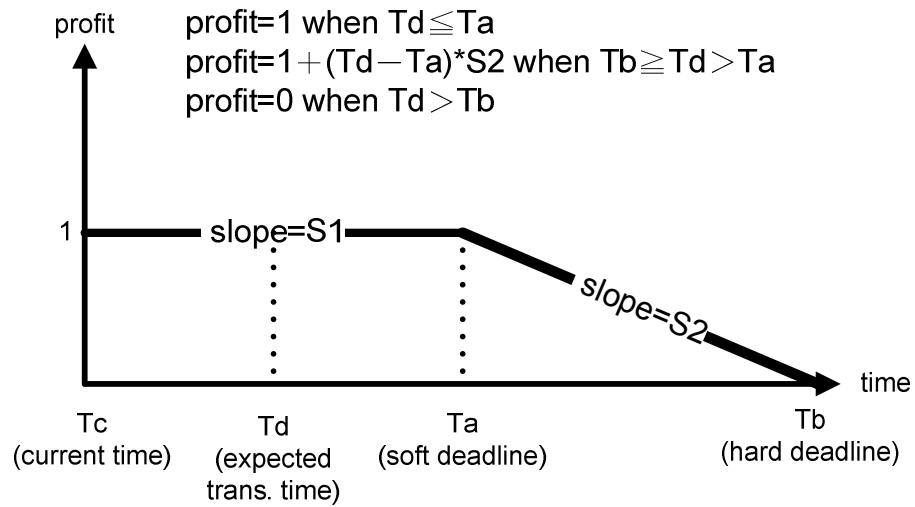


圖 3.4 Profit function Slope-Down

第二種 profit function 是 Slope-Down，profit function 在 soft deadline 之前斜率為 0，表示在 soft deadline 之前送出並無額外利益，但過了 soft deadline 斜率下降必須儘早送出封包否則利益受損，如過了 hard deadline 則無 profit，便將之丟棄。

C. Slope-Step :

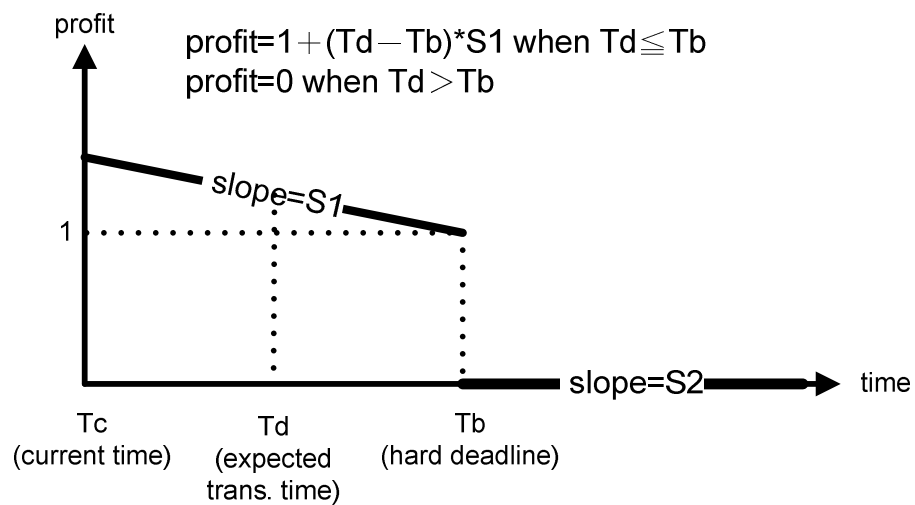


圖 3.5 Profit function Slope-Step

第三種 profit function 是 Slope-Step，當 router 在處理 Slope-Step type 的封包時，在 hard deadline 之前送出且越早送出 profit 越大。若無法在 hard deadline 之前送出則無 profit，便將之丟棄。

D. Double-Slope :

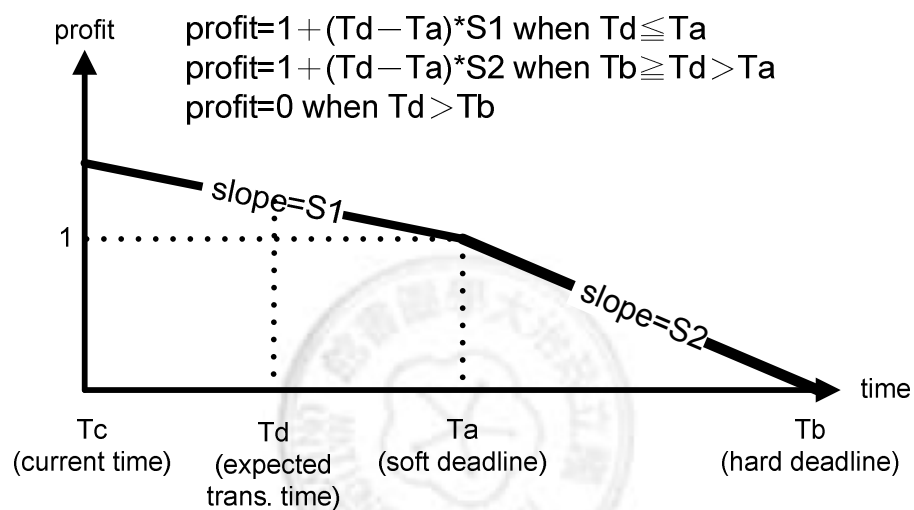


圖 3.6 Profit function Double-Slope

第四種 profit function 是 Double-Slope，當 router 處理 Double-Slope type 封包時，除了在 soft deadline 之前越早送出 profit 越高之外，且 profit function 在 soft deadline 前後斜率不同代表送出時間所獲之不同利益，類似 Slope-Down 所描述。

3.4 Scheduling Policy

我們將四種 profit function (Step-Down、Slope-Down、Slope-Step 及 Double-Slope)，分別賦予不同的服務等級，不同的搭配有不同的效果，每種組合我們稱之為 scheduling policy，網路管理者必須根據網路情況及自身的營運目標選擇最佳的 scheduling policy，以得到最佳利益。

3.5 Priority Multiple Queue router (PMQ)

論文[13]提出兩種 output queue router 架構。其中 Single Preemptive Queue router 架構雖然能有較佳的效能，不過在實作方面較不容易，僅能作為參考，實際上被採用之機會不大。因此又提出了 Multiple FIFO Queue router 架構以補足實作不易的缺點。

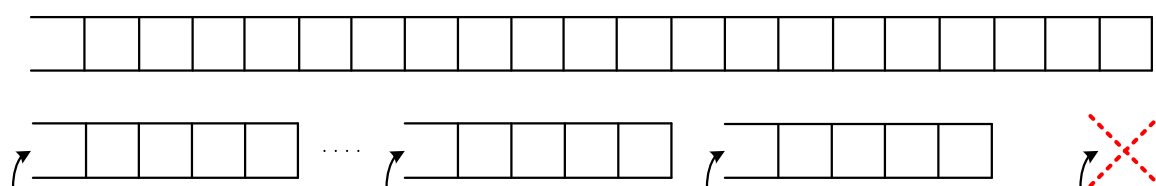


圖 3.7 Single Preemptive Queue 與 Multiple FIFO Queue router 的比較

Single Preemptive Queue router 架構之所以會有較佳的效能是因為 router 可以將進入的封包插到 output queue 的任意位置；而 Multiple FIFO Queue router 架構下個別的 output queue 是由數個 FIFO queue 所組成，僅能將進入的封包插入其中一個 queue 的尾端。

若將 Multiple FIFO Queue router 與 Single Preemptive Queue 架構比較，如圖 3.7 所示，可以發現可以插隊的位置大幅減少，而且缺少最前面的位置以供插隊，導致非常緊急的封包也要排隊影響排程效果。我們的目標就是改善 Multiple FIFO Queue router 架構，讓緊急的封包可以插在最前端送出。

3.5.1 PMQ 架構

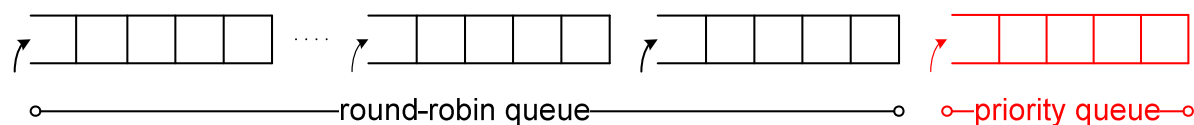


圖 3.8 Priority Multiple Queue 中的 Priority Queue

為了維持 Multiple FIFO Queue router 硬體容易實作的特性，我們一樣使用數個 FIFO queue，而讓其中一個 queue 的優先權提高以供較為緊急的封包插入，稱作 priority queue，如圖 3.8 所示。其他的 queue 稱作 round-robin queue。

圖 3.9 是 PMQ 的架構，pre-scheduler 依封包的 profit function 選擇適當的 queue 之尾端插入，使得新封包送出順序之總獲利在所有選擇中為最大；而 priority queue 只要有封包 post-scheduler 就會先送出，round-robin queue 則維持原本 MQ router 架構的作法，如同 2.1.2.2 所述。其中 scheduling database 包含某些排程資訊，例如 profit function、前瞻試排程相關資訊等。

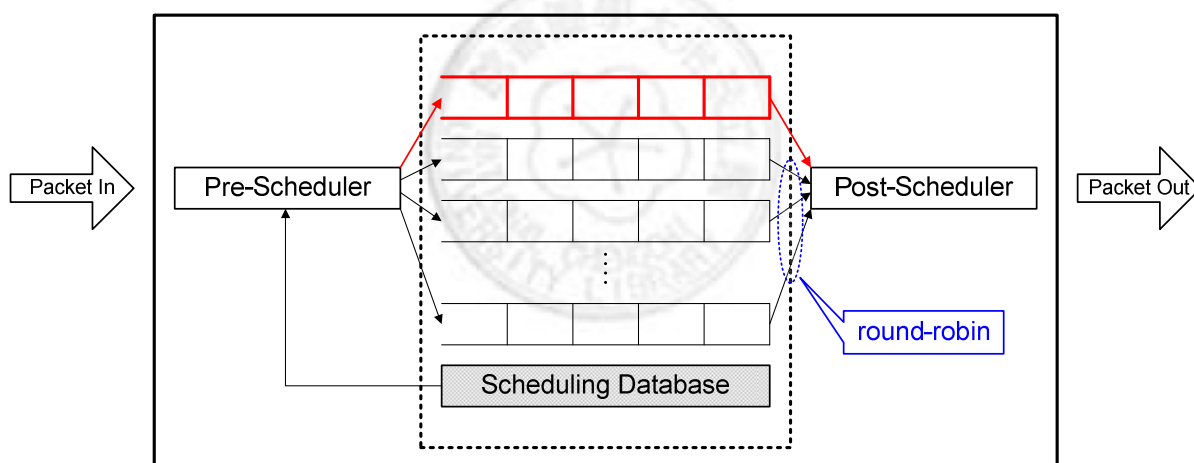


圖 3.9 Priority Multiple Queue router 架構

3.6 前瞻式排程 (Looking Ahead Scheduling)

本研究所提出的前瞻式排程相對於[13]所提出的獨立排程，最主要的差別是前瞻式排程中 router 會參考封包在後續路程上各 router 的狀態讓 scheduler 能更準確的預估封包及時到達目的地的機會。

3.6.1 設計理念及目標

我們必須找到一個估計封包在後續路程上各 router 狀態的方法，且用以做為動態調整 profit function 參數的依據，並找出最佳調整參數。前瞻式排程的精神在於讓各個 router 間互相分享資訊，反應網路狀況，以決定較佳的封包排程。

3.6.2 估計後續 router 負載

有數個方法可以估計後續 router 之負載，例如傳送端定期發送探測封包量測網路狀態，等接到由接收端回傳 ACK 所帶的資訊後，再用以估計後續 router 之負載；或查訊 router 中 queue 的封包傳送變化量，若是網路負載較重，則 queue 中封包的傳送變化量會較為緩慢。但是這些方法都會增加網路的負載或演算法的複雜度。

於是我們利用 routing table 中的資訊來估計後續 router 之負載，以進行前瞻式排程。假設 router 使用 Link-State routing protocol，在這類路由協定中，每個 router 都擁有相同且完整網路拓樸及 link cost 的資訊，Link-State routing protocol 提供了幾項 metrics 來計算 link cost，例如流量負載、可用頻寬或 delay time 等，而在本研究中將會假設 routing metric 為 delay time。

OSPF (Open Shortest Path First)

OSPF 是一種使用 Link-State 演算法的 routing protocol。在這個協定中，link cost 是從插入 queue 開始計算，所以包含 queue delay、transmission time 及 propagation delay，並且在每隔一段周期或拓樸有改變皆會更新 routing table 中的 link cost 資訊，如此能反映網路實際狀況及負載。

由於本法使用 routing table 中的資訊，所以效能與 routing protocol 更新資訊的速度極為相關，如前述中週期性的更新 routing table 資訊，理論上越頻繁會對我們的方法越

好，越能彰顯我們前瞻式排程的效能；但過多的 routing 封包亦可能降低整體網路效能。但 routing protocol 不在我們的研究範圍內。

3.6.3 動態調整 profit function 參數

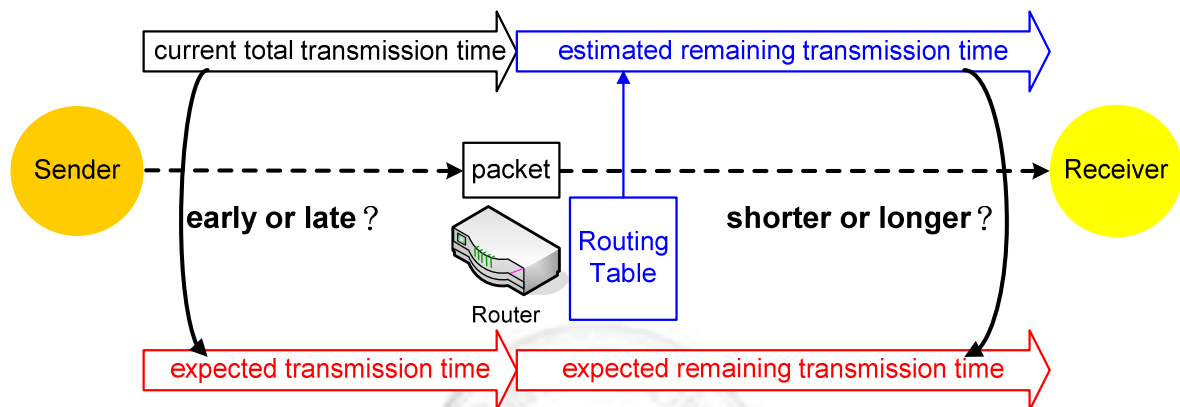


圖 3.10 動態調整 profit function 參數的依據

本研究假設每個封包在傳遞間所經歷的網路元件都有事先預定的停留時間，因此可以預定封包到達每一個 router 的時間及其送出時間。而由於 IP 封包的傳遞時間起伏不定，會使得封包已傳遞時間比預定的時間早到或遲到。

另外當封包到達每個 router 時，我們可以透過 routing table 中的資訊預估該封包的剩餘傳遞時間比預定剩餘時間長或短。我們可以利用這些資訊來動態調整 profit function 的參數，使得封包的緊急程度不僅決定於目前所耗用時間之狀況，更根據未來可能的時間消耗狀況決定其緊急程度，如圖 3.10 所示。

圖 3.11 為封包在傳遞過程中的幾種可能狀況。其中最緊急的狀況即為封包實際到達時間比預定到達時間晚，且預估後續的傳遞時間又比預定剩餘時間長，這類封包我們必須儘可能提前送出以求適時到達目的地。以此類推其他狀況及其處理方式。

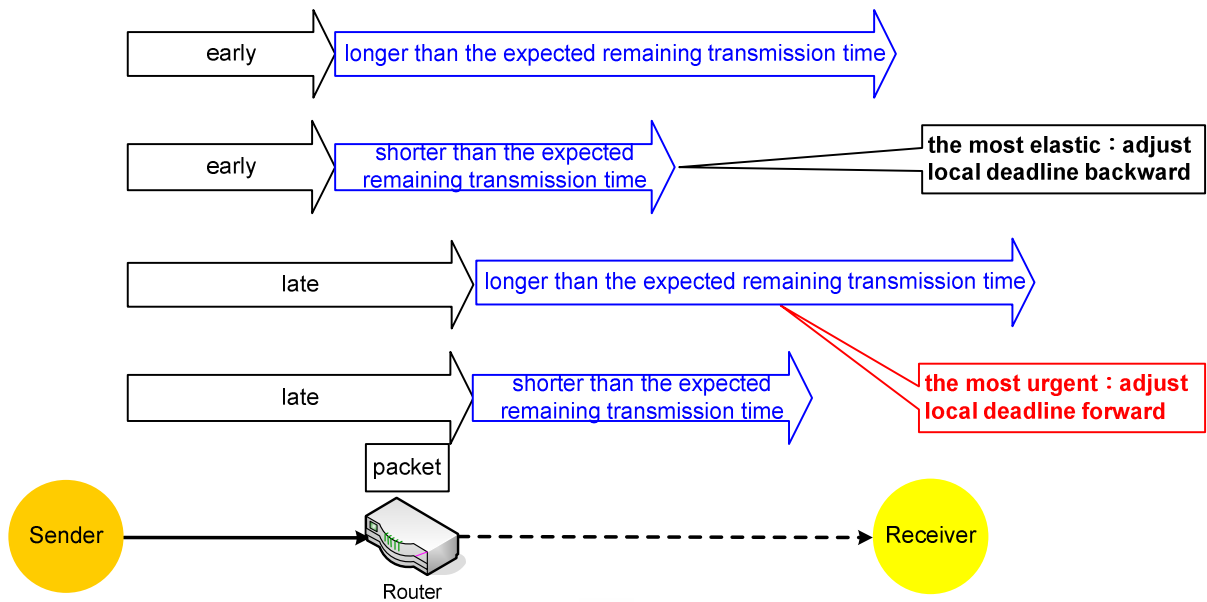


圖 3.11 封包傳遞中的幾種狀況

[13]的四種 profit function，其參數的設定是採用較為簡單的方式。如 Step-Down 與 Slope-Step 是將 hard deadline 設為其預定送出時間，而 Slope-Down 與 Double-Slope 將 soft deadline 設為其預定送出時間，並將 hard deadline 適度延長一段時間。

表 3.3 是 local deadline 調整公式的參數。在前瞻式排程中，我們將實際已傳遞時間減去預定已傳遞時間作為該封包在傳遞中的已拖延時間，而將估計剩餘傳遞時間減去預定傳遞剩餘時間作為後續預計延遲的時間，兩者相加結果即為封包在整段傳遞過程中可能的預計總拖延時間。若預估該封包落後預定時程，就將 local deadline 適當地往前調(亦即提高其優先度)，使 scheduler 將封包提早送出。

相反的，若預估結果認為封包在整段傳遞過程中可能提前到達目的地，我們就可以將 local deadline 適當地往後調使封包稍緩送出，scheduler 便可以藉此補救其他較緊急的封包。

我們另外設定一個調整係數調整各 router 的 local deadline 以逐步調整封包的傳遞時間。我們將會以實驗找出最佳的調整係數。

表 3.3 調整 local deadline 的參數及定義

Notation	Name	Definition
d_{pre}	已拖延的時間 (current exceeded transmission time)	實際已傳遞時間 - 預定已傳遞時間 (total transmission time - expected transmission time)
d_{post}	後續預計延遲的時間 (estimated exceeding transmission time afterward)	估計剩餘傳遞時間 - 預定傳遞剩餘時間 (estimated remaining transmission time - expected remaining transmission time)
$d_{pre} + d_{post}$	預計總拖延時間 (estimated total exceeding transmission time)	已拖延的時間 + 後續預計延遲的時間 (current exceeded transmission time + estimated exceeding transmission time afterward)
c	調整係數 (adjusting coefficient)	介於 0 與 1 之間的實驗值，用以控制調整 profit function 的程度 (an empirical value between 0 and 1 for controlling the level of profit function adjustment)

表 3.4 是 local deadline 的調整公式，我們將預計總拖延時間乘上調整係數，Step-Down 與 Slope-Step 的 hard local deadline 減去該乘積，而 Slope-Down 與 Double-Slope

則是 soft local deadline 減去該乘積。如此調整後的 local deadline 可以反應封包在整段傳遞過程中可能的預計總延遲時間。

表 3.4 Local deadline 調整公式

Profit function	Adjusting formula
Step-Down Slope-Down	hard local deadline - = $c \cdot (d_{pre} + d_{post})$
Slope-Down Double-Slope	soft local deadline - = $c \cdot (d_{pre} + d_{post})$

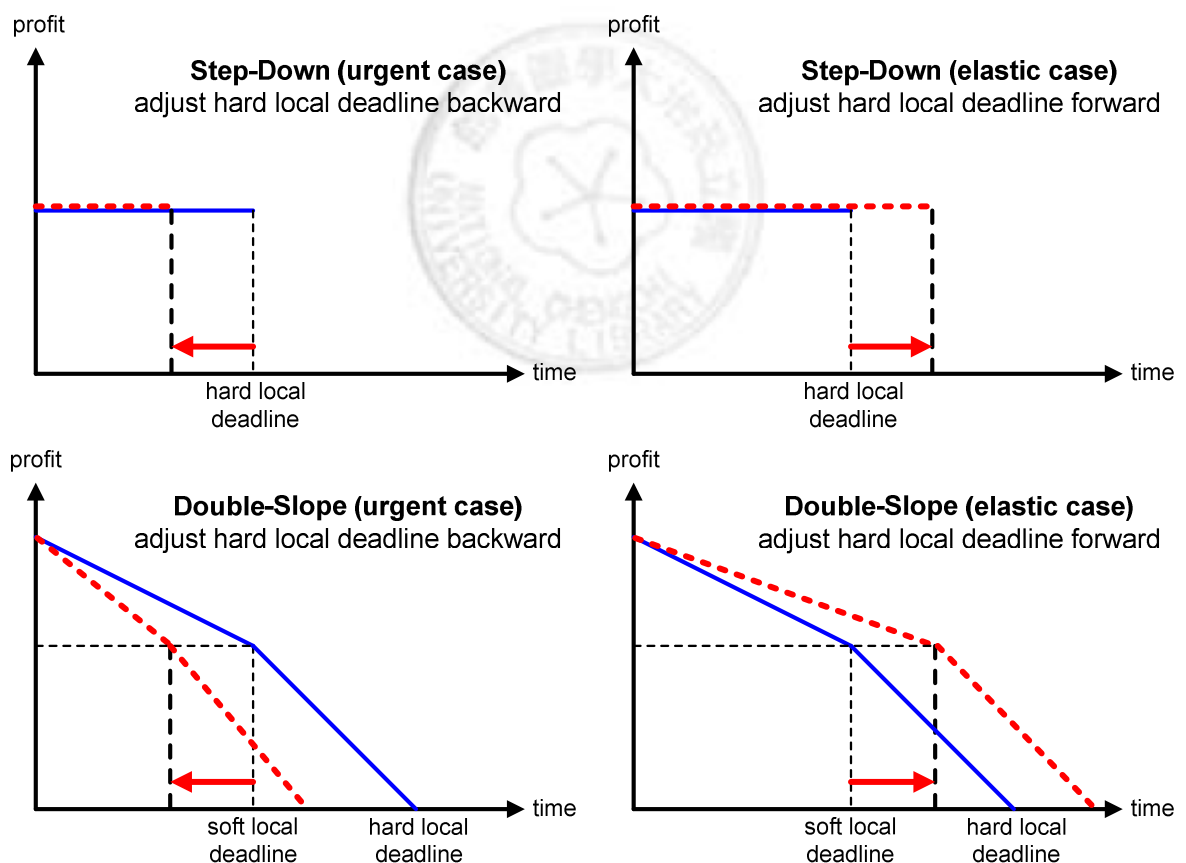


圖 3.12 調整 profit function 的 local deadline

圖 3.12 是我們針對不同 profit function 調整 local deadline 的示意圖。不同 profit function 會有些許差異，例如 Step-Down 調整後 profit rate 維持不變，而 Double-Slope 在調整後，其 profit rate 會隨著 hard /soft deadline 改變，如此 scheduler 更能根據封包的緊急程度決定封包送出的順序以最大化總獲利和。

所有以 deadline 為基礎的研究都會有時間同步的問題，在我們的實驗中若有 router 時間不同步的情況發生，router 在處理某些時間較接近 deadline 的封包，可能會發生捨棄不該捨棄的封包或不捨棄該捨棄的封包之情況；但對於其他封包較無影響。所以我們僅能假設相鄰 router 的最大誤差時間不足以影響我們的排程演算法。

3.7 LAS 排程演算法

圖 3.8 是 Priority Multiple Queue router 的架構。在此種架構下我們尚需設計前端排程器 (pre-scheduler) 與後端排程器 (post-scheduler)，而排程演算法則是設計在 pre-scheduler 與 post-scheduler 上面。pre-scheduler 依封包的 profit function 選擇適當的 queue 插入尾端使封包送出順序獲得最大總獲利。而 priority queue 中只要有封包 post-scheduler 就會先送出；而對於其他 round-robin queue 依舊是採用輪詢(round-robin)的方式送出封包。

我們方法是利用 profit function 的方式表達進入一個 router 的封包的緊急程度及服務等級，而在前瞻式排程下 profit function 參數會依網路後續負載調整，再配合排程演算法來決定封包排程順序，以最大化總獲利和為目標以提高整體服務品質。以下是排程演算法的輸入與輸出：

輸入：數個 FIFO queue 所組成的 output queue，每個進入 router 的封包帶有給定的 profit function 及時間參數。

輸出：一個適當的插入位置(某條 queue 的尾端)，使封包送出順序可以有最大獲益。

3.7.1 演算法流程

圖 3.13 是 PMQ 架構中 pre-scheduler 的排程演算法流程圖，其中虛線方框部份為前瞻式排程。

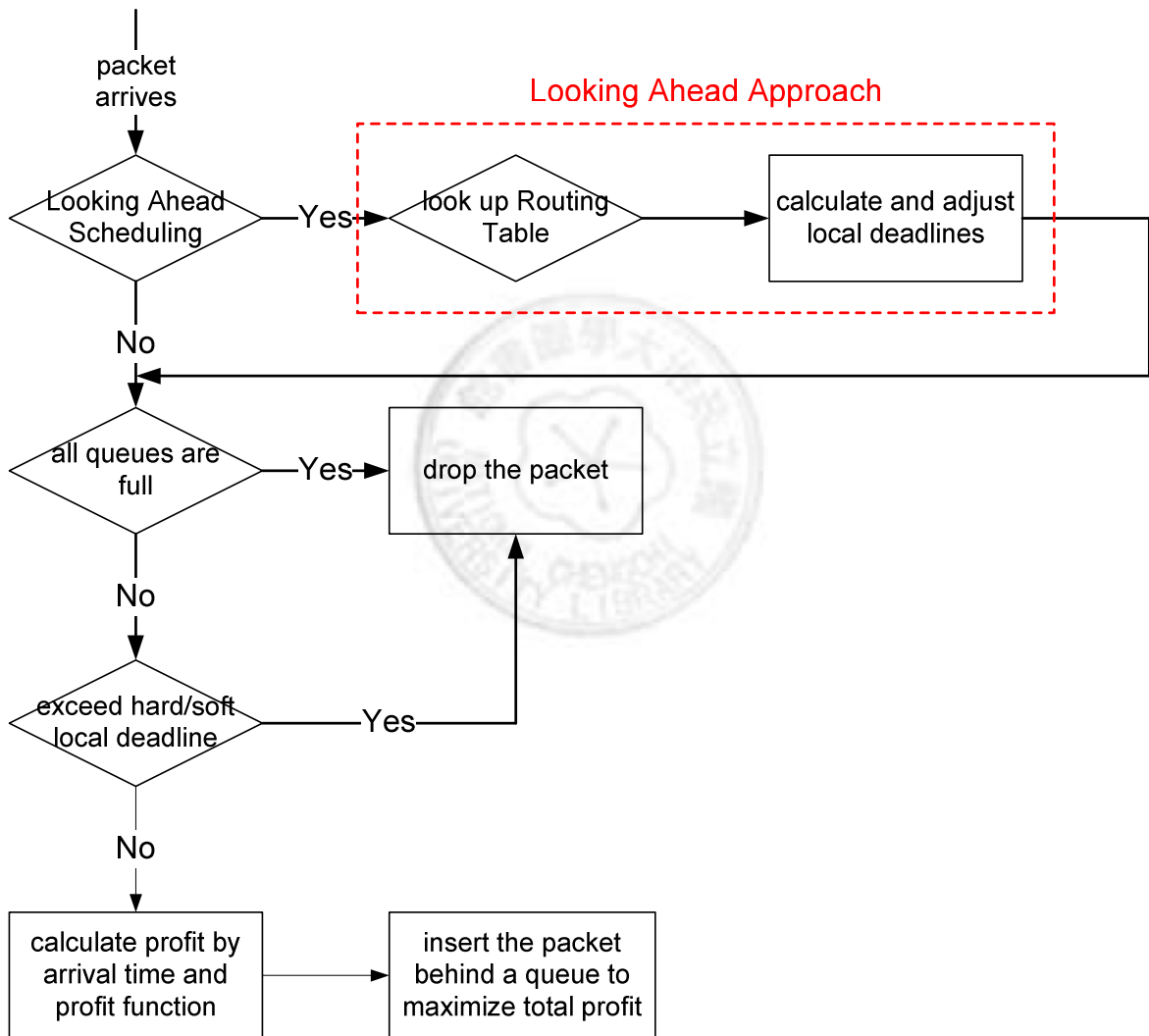


圖 3.13 Pre-scheduler 排程演算法流程圖

圖 3.14 是 PMQ 架構中 pre-scheduler 的排程演算法流程圖，其中虛線方框部份為 round-robin queue。

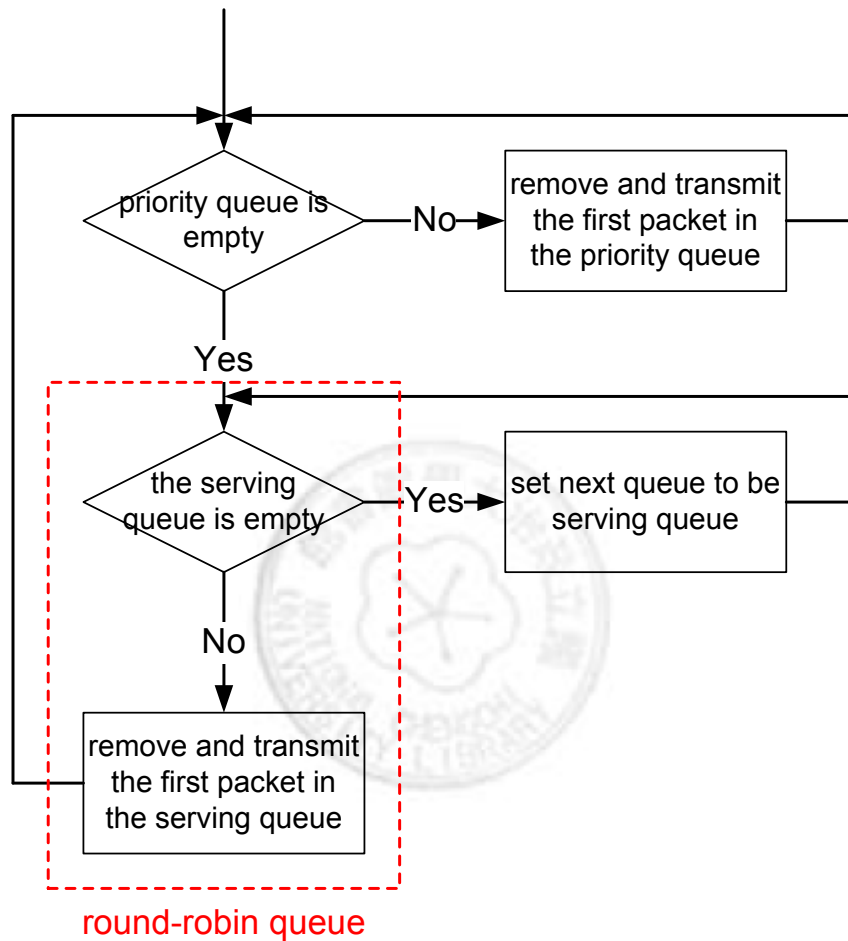


圖 3.14 Post-scheduler 排程演算法流程圖

3.7.2 演算法虛擬碼 (Pseudo code)

```
// pre-scheduler
Enqueue ( Packet , ArrivalTime)
{
  if ( IsLookingAhead )
  {
    LookUp ( RoutingTable ) ;
    Adjust ( Hard/SoftDeadline) ;
  }
  if ( IsFull ( AllQueue ) ) Drop(Packet) ;

  if ( ArrivalTime > HardDeadline ) Drop(Packet) ;
  else
  {
    ProfitCalculate ( PacketProfitFunction , ArrivalTime) ;
    for (AllQueue)
    {
      FindMaxTotalProfit ; //find a queue to insert
    }
    Queue → Enqueue ;
  }
}

// post-scheduler
Deque ( )
{
  if ( IsEmpty ( PriorityQueue ) )
  {
    // round-robin queue
    if ( IsEmpty ( ServingQueue ) ) NextRoundRobinQueue ← ServingQueue ;
    // remove and transmit the first packet in the serving queue
    else ServingQueue → Dequeue ;
  }
  else PriorityQueue → Dequeue ;
}
```

第四章

效能評估

本研究以多組實驗來評估我們提出的方法，藉由在接收端觀察數個效能指標了解我們方法的效能，並且與[13]中所提出的方法比較。

4.1 實驗環境

NS-2[25]是一套模擬 IP 網路的模擬平台，NS-2 (Network Simulator - version 2)。

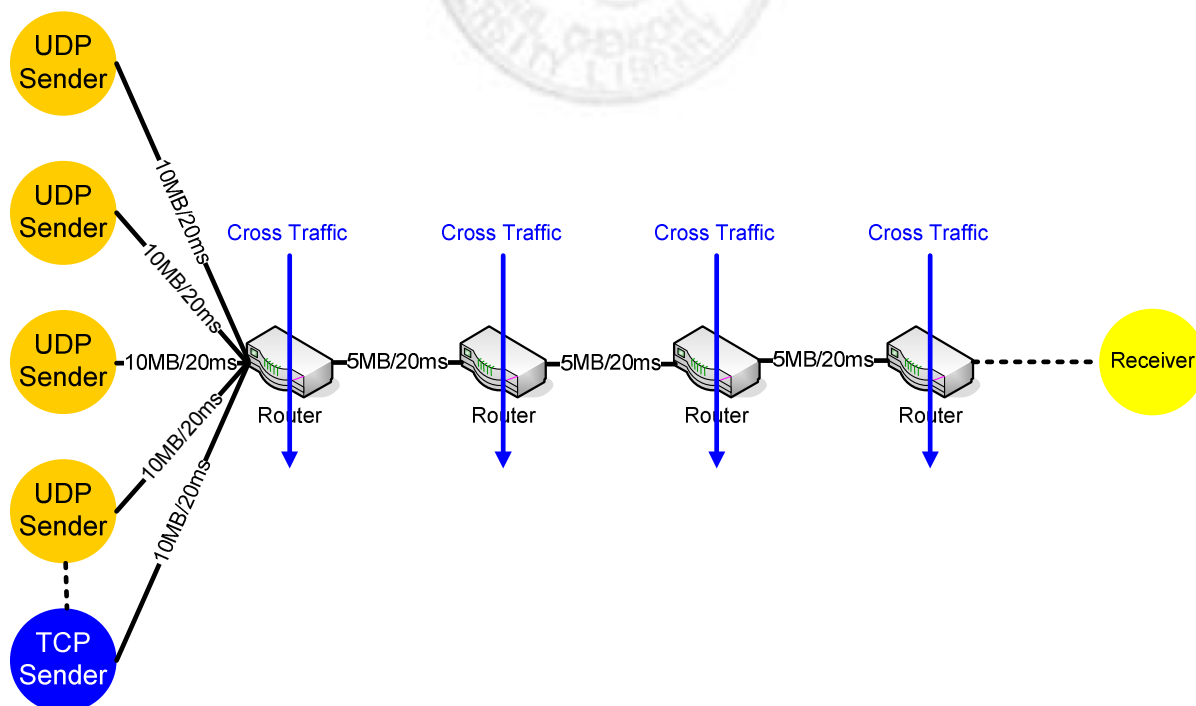


圖 4.1 實驗拓模一

圖 4.1 是我們大部分實驗的拓樸，其中虛線部分代表實驗控制變數。在傳送端我們設定 real-time 以及 Background class traffic，每個 router 都會有 cross traffic 經過，cross traffic 採用 Background class traffic，其中 real-time class traffic 以 CBR 的方式傳送，而 Background class traffic 以 FTP 的方式傳送。

4.2 實驗設計

我們變動網路負載、路徑長度及 queue size，觀察不同網路負載與不同網路環境時的效能表現。

而當不同的服務等級的封包進入 router 而資源不足時，必須給予差異化待遇，讓資源依優先等級調配，如此可以提高整體滿意度。在我們的架構中，所能著力之處在於依照封包的服務等級選擇一個適當 profit function 造成差異化待遇。研究的重點即在於如何調配 profit function 以獲得最大效能。由於理論分析的困難，我們將藉由 NS-2 模擬實驗找出適當的配方。

如果網路資源足夠，不同的服務等級的封包可以並存，沒有資源排擠的問題，於是在設計實驗時，特地把網路流量增大，觀察排擠服務等級較低的封包後，real-time 服務等級的封包間資源搶奪的行為。

我們先以單一 TSCO 等級作實驗，研究各種 profit function 的行為，再以多個 TSCO 等級服務進行實驗。

A. Single TSCO class and Background class

觀察使用不同 router 架構及排程演算法的效能表現。

B. Multiple TSCO class and Background class

Conversational 及 Streaming 兩種服務等級的 real-time traffic 混合傳送，觀察多 TSCO

等級服務的封包調配不同 profit function 間的資源搶奪行為與不同等級服務間的差別化待遇。這部份是我們研究的重點，我們將藉由實驗找出適合的 policy。

4.2.1 實驗步驟

我們以三個步驟進行實驗，如圖 4.2 所示。論文[13]提出兩種針對不同架構的排程演算法，並在獨立排程上進行；步驟 1 先觀察我們所提出的 PMQ 架構，步驟 2 觀察前瞻式排程演算法的效能表現，並找出最佳調整參數；最後再以步驟 3 的多服務等級實驗找出最佳的 profit function 組合。

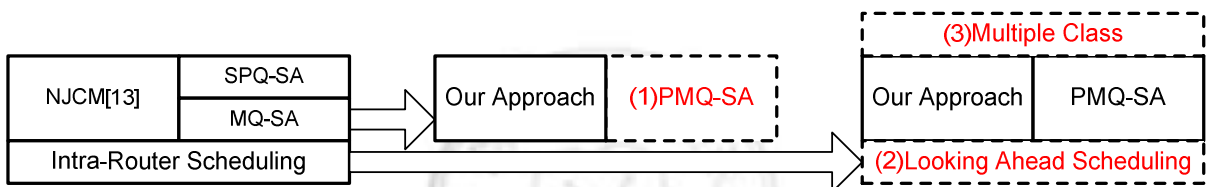


圖 4.2 實驗步驟

1. 在單一服務等級環境中比較 IRS-PMQ 與 IRS-MQ

觀察改善 Multiple FIFO Queue router 架構的效能。

2. 在單一服務等級環境中比較 LAS-PMQ 與 IRS-PMQ

觀察前瞻式排程動態調整 deadline 的效能並找出最佳調整參數。

3. 多服務等級效能評估實驗

觀察多服務等級的封包調配不同 profit function 間的行為與不同等級服務間的差異化處理。最後與 Simulated Priority Queue 排程演算法[15]做效能之比較。

4.2.2 實驗參數

實驗所用的參數值，如表 4.1 所示。

表 4.1 實驗參數

Parameter	Range
Number of CBR flows	Specified in individual experiments
Data rate of CBR flows	
Queue size	
Hop count	
Router architecture	FIFO,SPQ,MQ,PMQ,PQ
Scheduling algorithm	IRS,LAS
Profit function	Step-Down, Slope-Down, Slope-Step, Double Slope

4.3 評估指標

A. 單一服務等級實驗評估指標

觀察 jitter 平均值、封包遺失率及封包平均延遲時間的變化作為設計多服務等級 scheduler 之參考。

B. 多服務等級實驗評估指標

我們以公式(3.1)所定義的 total charge 來做為多服務等級實驗評估指標。

4.4 實驗一：在單一服務等級環境中比較 IRS-PMQ 與 IRS-MQ

本實驗分別調整網路負載、路徑長度及 queue size 以觀察我們提出的 PMQ router 架構的效能變化(jitter 平均值、封包遺失率、平均延遲時間)。在實驗一我們會放入傳統 FIFO

與[13]中的 SPQ router 架構的實驗結果以供比較。

在單一服務等級的實驗中，我們觀察賦予不同 profit function 的結果，發現僅存在一種 profit function 的狀況下差異並不大，差異化處理要在多種 profit function 共存才會顯示出來。所以實驗一我們僅放入 Step-Down 的實驗結果以供參考，以便觀察不同 router 架構之效能。

4.4.1 實驗 1A：探討負載對效能的影響

4.4.1.1 實驗目標

本實驗調整網路負載以比較 PMQ 與 MQ router 架構的整體效能變化。

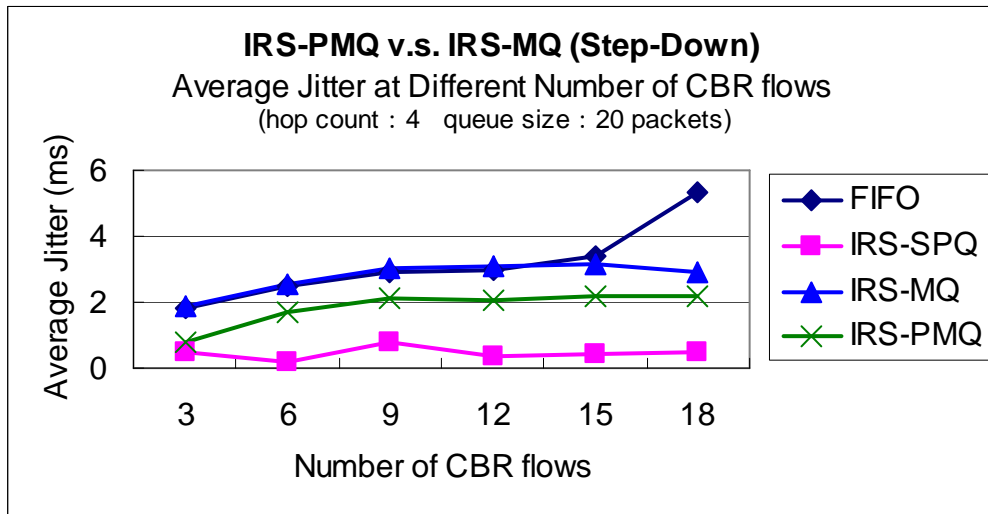
4.4.1.2 實驗流程

圖 4.1 是實驗 1A 的實驗拓樸。在這個實驗中逐漸增加即時性訊務個數以觀察不同網路負載的情形。

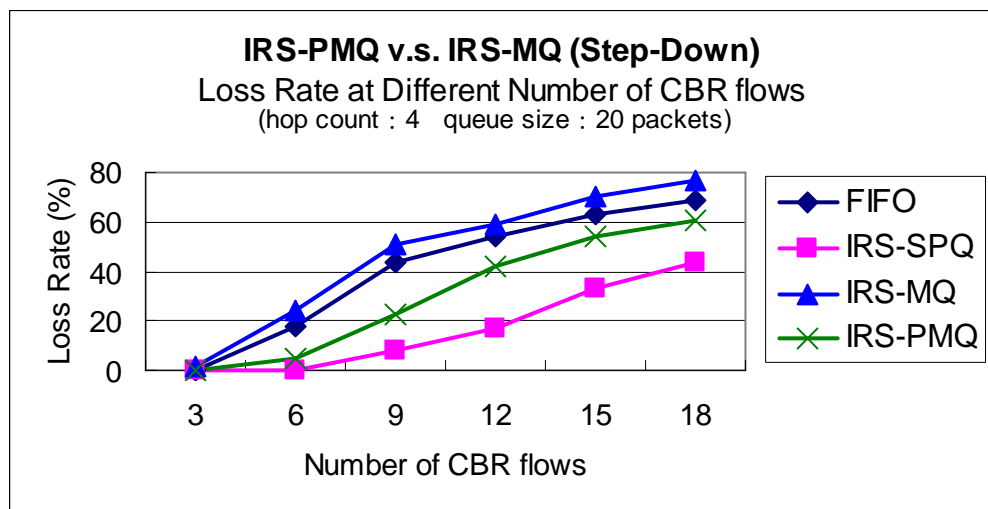
表 4.2 實驗 1A 參數

Parameter	Range
Number of CBR flows	3,6,9,12,15,18
Hop count	4
Queue size	20 packets

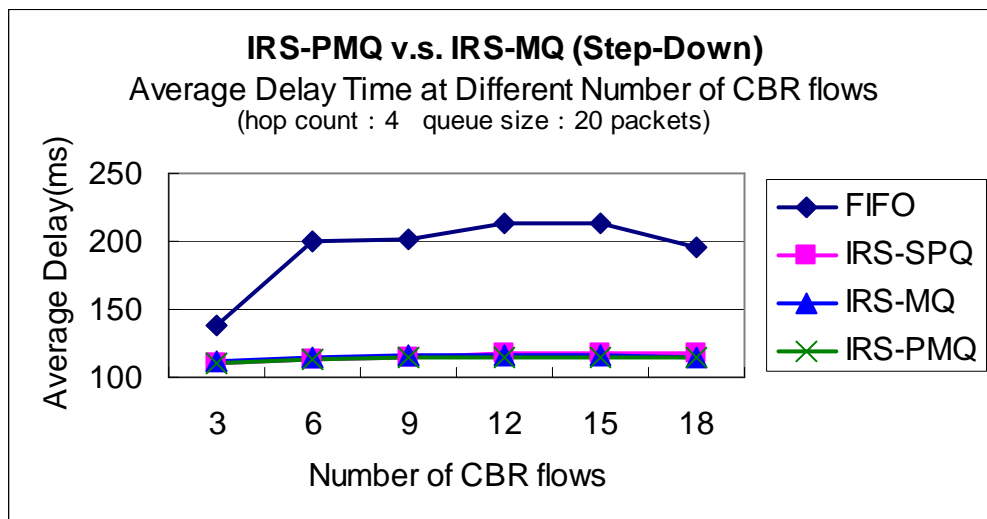
4.4.1.3 實驗結果與分析



(a)



(b)



(c)

圖 4.3 實驗 1A 變動負載下 IRS-PMQ 與 IRS-MQ 之效能比較

(a)Jitter 平均值 (b)封包遺失率 (c)平均延遲時間

圖 4.3 是實驗 1A 的結果。由實驗結果可以看出，在網路負載越來越大的情況下，我們所提出的 PMQ router 架構的確比 MQ router 架構更可以改善整體 QoS，尤其以 jitter 與封包遺失率更為明顯，更逼近 SPQ 排程演算法。

4.4.2 實驗 1B：探討路徑長度對效能的影響

4.4.2.1 實驗目標

本實驗調整路徑長度並觀察比較 PMQ 與 MQ router 架構的整體效能變化。

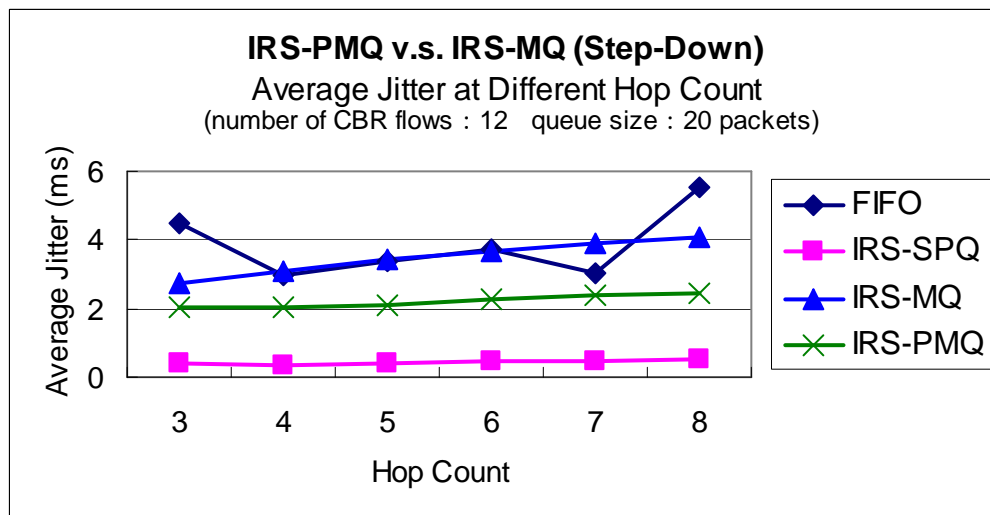
4.4.2.2 實驗流程

與實驗 1A 一樣，圖 4.1 是實驗 1B 的實驗拓樸，在這個實驗中漸漸增加 router 數量，以觀察在不同路徑長度下，PMQ router 架構的效能表現。

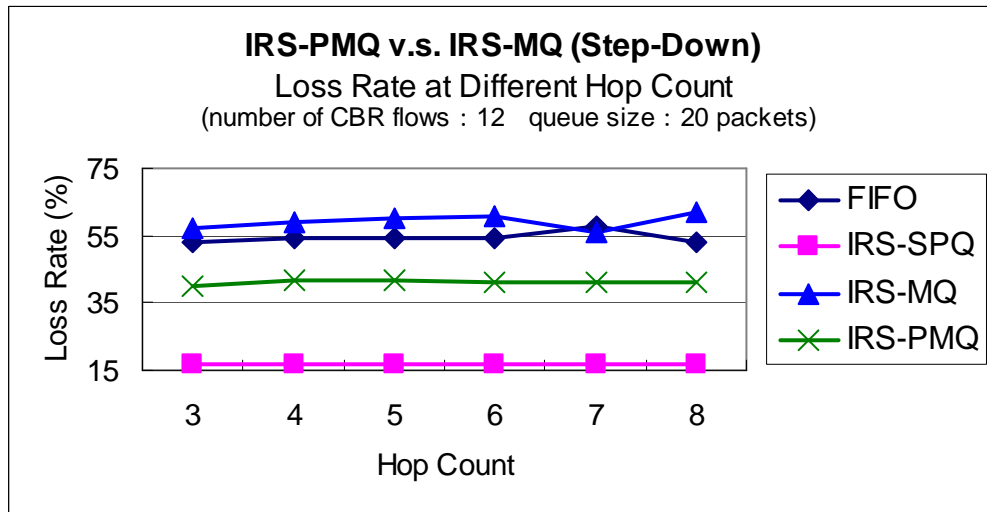
表 4.3 實驗 1B 參數

Parameter	Range
Number of CBR flows	12
Hop count	3,4,5,6,7,8
Queue size	20 packets

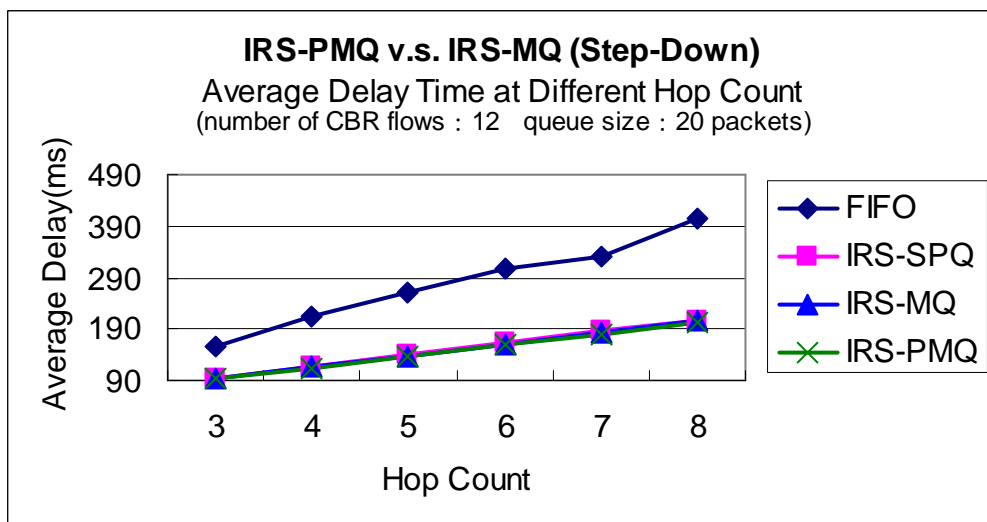
4.4.2.3 實驗結果與分析



(a)



(b)



(c)

圖 4.4 實驗 1B 變動路徑長度下 IRS-PMQ 與 IRS-MQ 之效能比較

(a)Jitter 平均值 (b)封包遺失率 (c)平均延遲時間

圖 4.4 是實驗 1B 的結果。由實驗結果可以看出，在路徑越來越長的情況下，我們提出的 PMQ router 架構一樣是都有不錯的效能，其中 jitter 相當穩定，並不會因為路徑變長而不穩定，封包遺失率也有不錯的表現。反觀 MQ router 架構下的 jitter 較為不穩定，而封包遺失率甚至不優於傳統 FIFO。

4.4.3 實驗 1C：探討 queue size 對效能的影響

4.4.3.1 實驗目標

本實驗調整 queue size 並觀察比較 PMQ 與 MQ router 架構的整體效能變化。

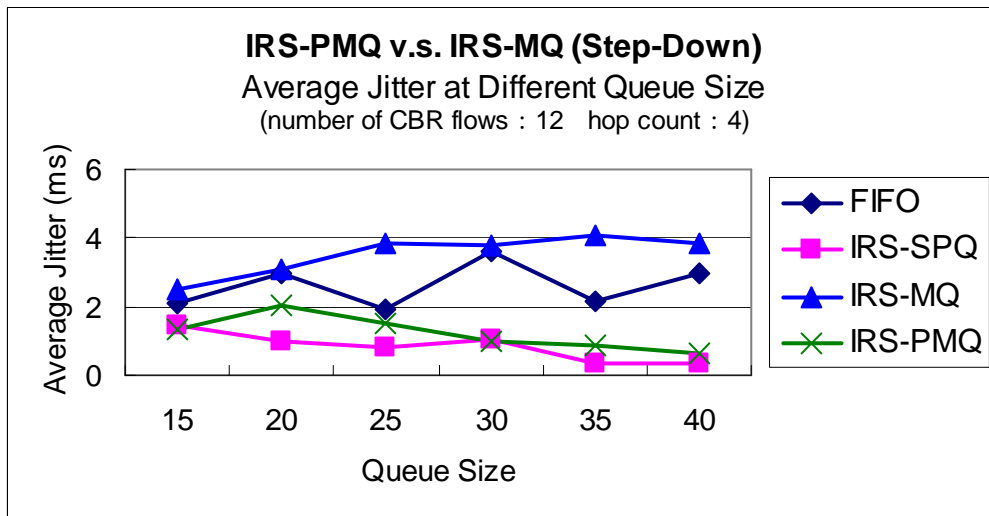
4.4.3.2 實驗流程

與實驗 1A 一樣，圖 4.1 是實驗 1C 的實驗拓樸，在這個實驗中觀察在 queue size 不同的情況下，PMQ router 架構的效能表現。

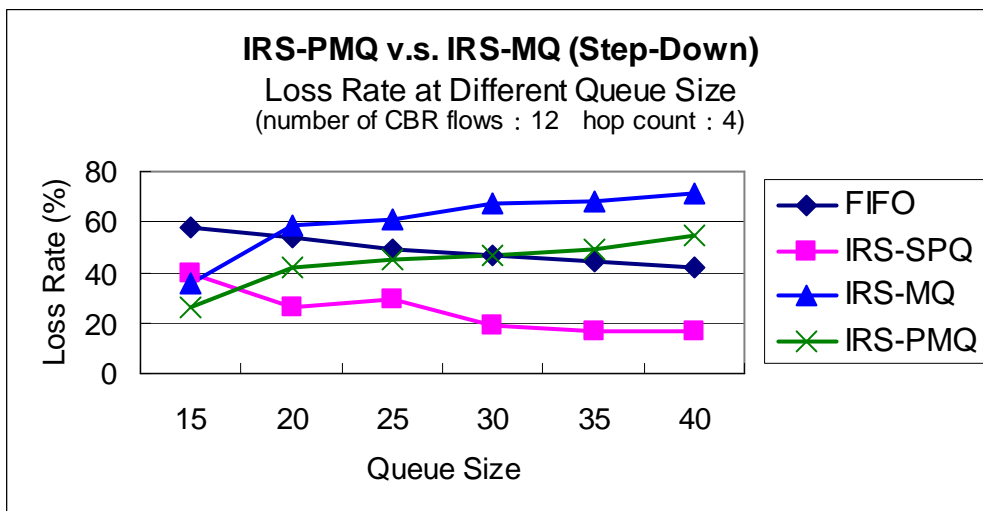
表 4.4 實驗 1C 參數

Parameter	Range
Number of CBR flows	12
Hop count	4
Queue size	15,20,25,30,35,40

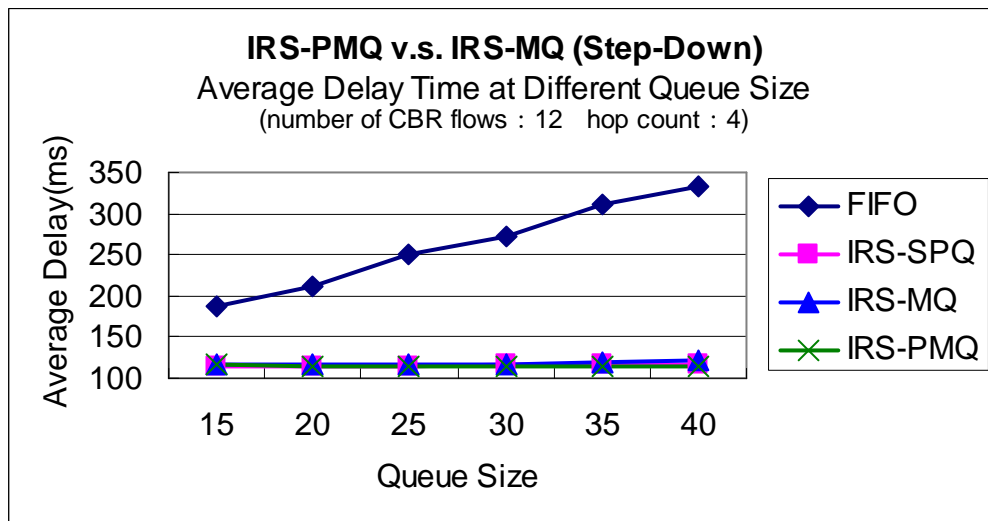
4.4.3.3 實驗結果與分析



(a)



(b)



(c)

圖 4.5 實驗 1C 變動 Queue Size 下 IRS-PMQ 與 IRS-MQ 之效能比較

(a)Jitter 平均值 (b)封包遺失率 (c)平均延遲時間

圖 4.5 是實驗 1C 的結果。由實驗結果可以看出，在 queue size 越來越大的情況下，[13] 中的 MQ router 架構的 jitter 及封包遺失率會越來越高，因為 queue size 越大則一些較緊急的封包在其中單一 queue 裡排隊的時間會變長導致效能不佳。而 PMQ router 架構一樣是都有不錯的效能，其中 jitter 相當穩定，甚至可以逼近 SPQ router 架構，並不會因為 queue size 變大而變差。但封包遺失率受限於 multiple FIFO queue router 架構的特性表現較差，但仍在可接受的範圍。

4.5 實驗二：在單一服務等級環境中比較 LAS-PMQ 與 IRS-PMQ

本實驗分別調整網路負載、路徑長度及 queue size 以觀察我們提出的前瞻式排程的整體效能(Jitter 平均值、封包遺失率、平均延遲時間)。在實驗二中，我們與 PMQ router 架構下的獨立排程演算法做比較，並且藉以找出最佳的調整係數。

4.5.1 實驗 2A：探討負載對效能的影響

4.5.1.1 實驗目標

本實驗調整網路負載並觀察前瞻式排程演算法的整體效能變化。

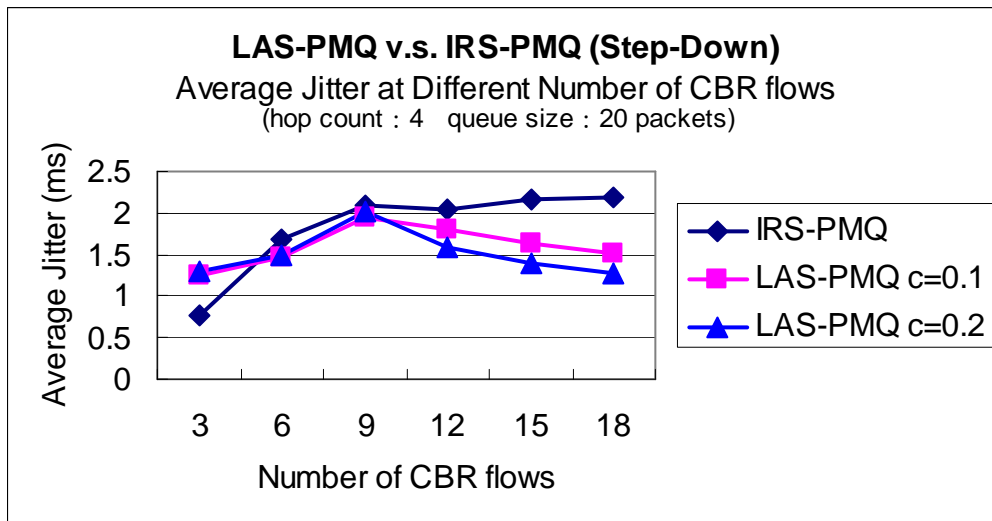
4.5.1.2 實驗流程

圖 4.2 是我們的實驗拓樸。在這個實驗中逐漸增加即時性訊務個數以觀察不同網路負載的情形。

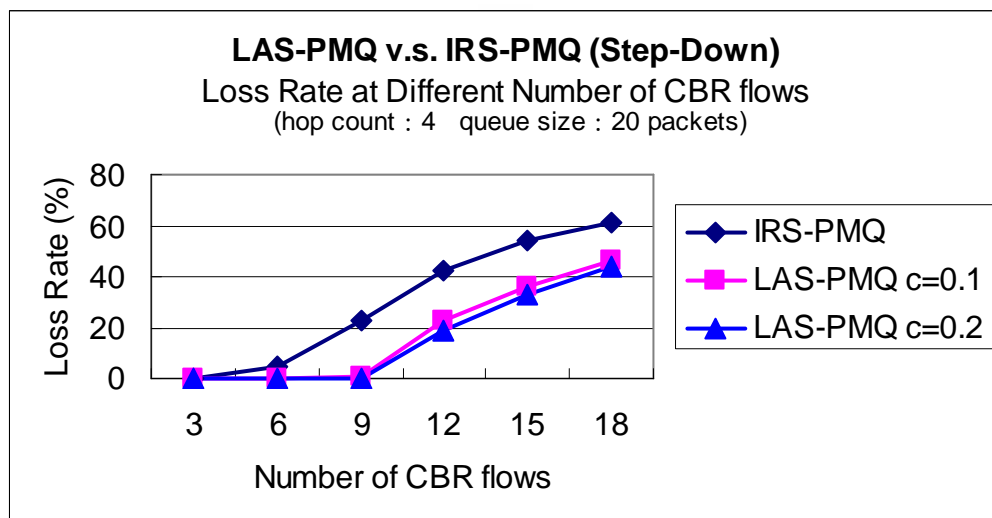
表 4.5 實驗 2A 參數

Parameter	Range
Number of CBR flows	3,6,9,12,15,18
Hop count	4
Queue size	20 packets

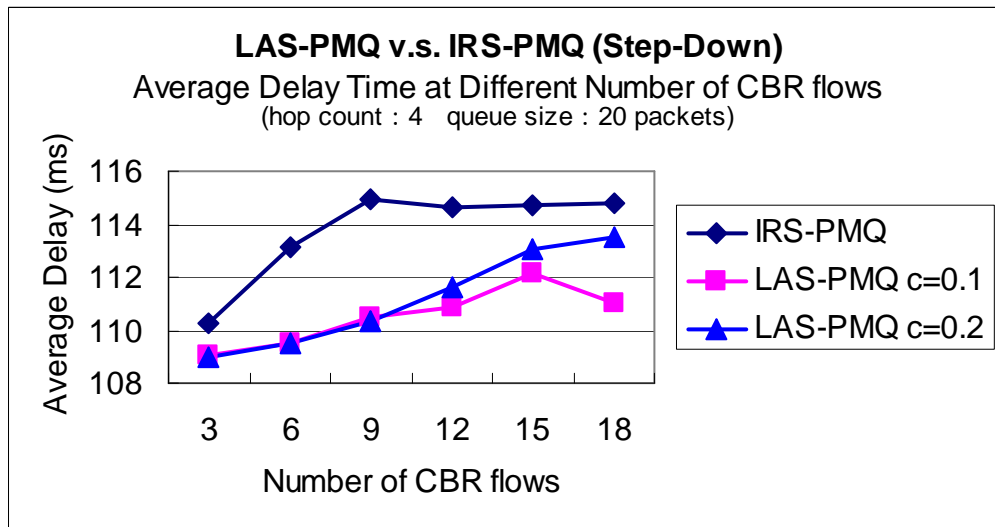
4.5.1.3 實驗結果與分析



(a)



(b)



(c)

圖 4.6 實驗 2A 變動負載下 LAS-PMQ 與 IRS-PMQ 之效能比較

(a)Jitter 平均值 (b)封包遺失率 (c)平均延遲時間

圖 4.6 是實驗 2A 的結果。由實驗結果可以看出，在網路負載越來越大的情況下，前瞻式排程演算法的 jitter 與平均延遲時間都小有改善，尤其封包遺失率的改善更為顯著。

4.5.2 實驗 2B：探討路徑長度對效能的影響

4.5.2.1 實驗目標

本實驗調整路徑長度並觀察前瞻式排程演算法的整體效能變化。

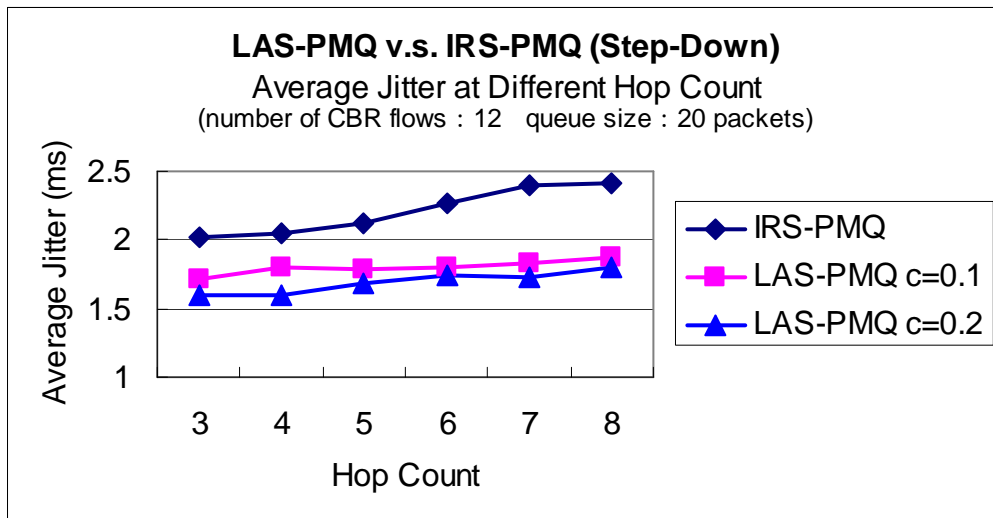
4.5.2.2 實驗流程

與實驗 2A 一樣，圖 4.1 是實驗 2B 的網路拓樸。在這個實驗中漸漸增加 router 數量，以觀察在路徑越來越長的情況下，前瞻式排程演算法的效能表現。

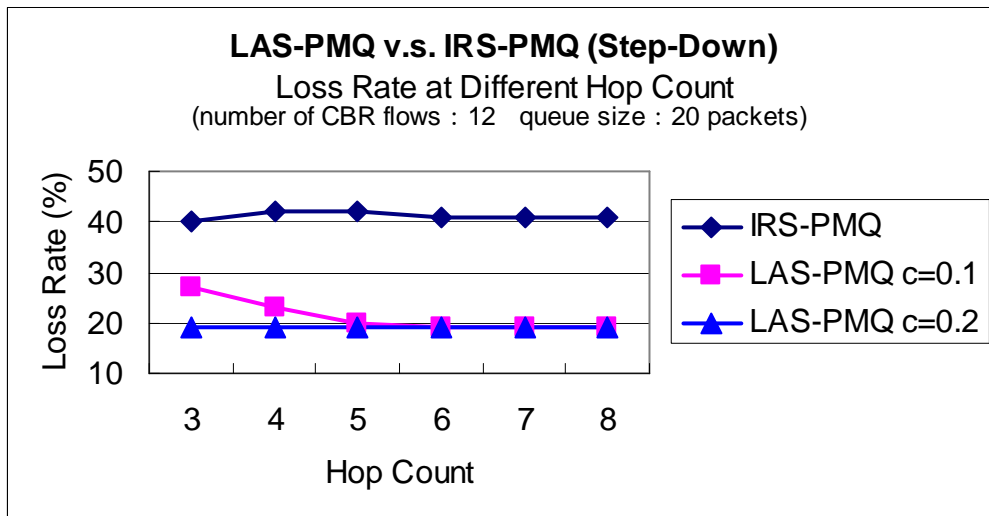
表 4.6 實驗 2B 參數

Parameter	Range
Number of CBR flows	12
Hop count	3,4,5,6,7,8
Queue size	20 packets

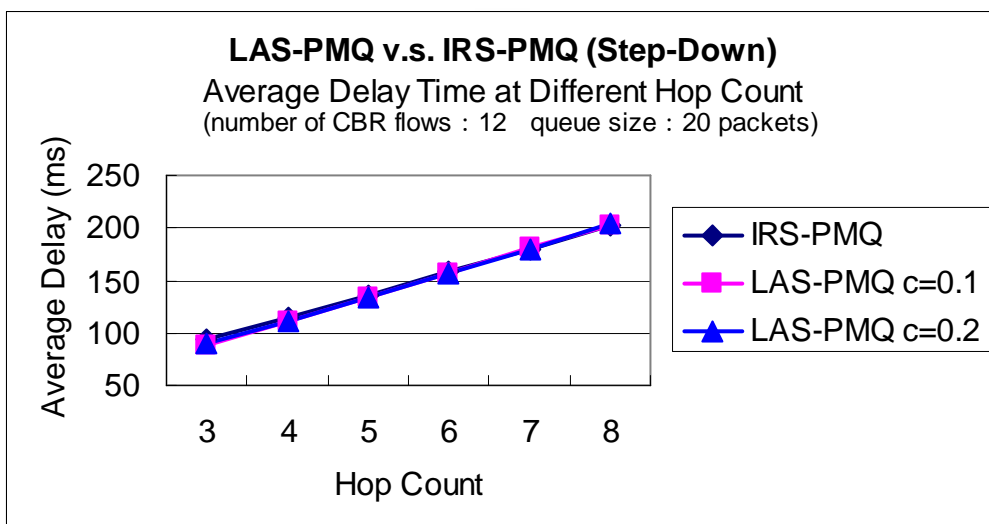
4.5.2.3 實驗結果與分析



(a)



(b)



(c)

圖 4.7 實驗 2B 變動路徑長度下 LAS-PMQ 與 IRS-PMQ 之效能比較

(a)Jitter 平均值 (b)封包遺失率 (c)平均延遲時間

圖 4.7 是實驗 2B 的結果。由實驗結果可以看出，在路徑越來越長的情況下，前瞻式排程演算法的 jitter 可以有小幅改善，而封包遺失率的改善幅度更為明顯。

4.5.3 實驗 2C：探討 queue size 對效能的影響

4.5.3.1 實驗目標

本實驗調整 queue size 並觀察前瞻式排程演算法的整體效能變化。

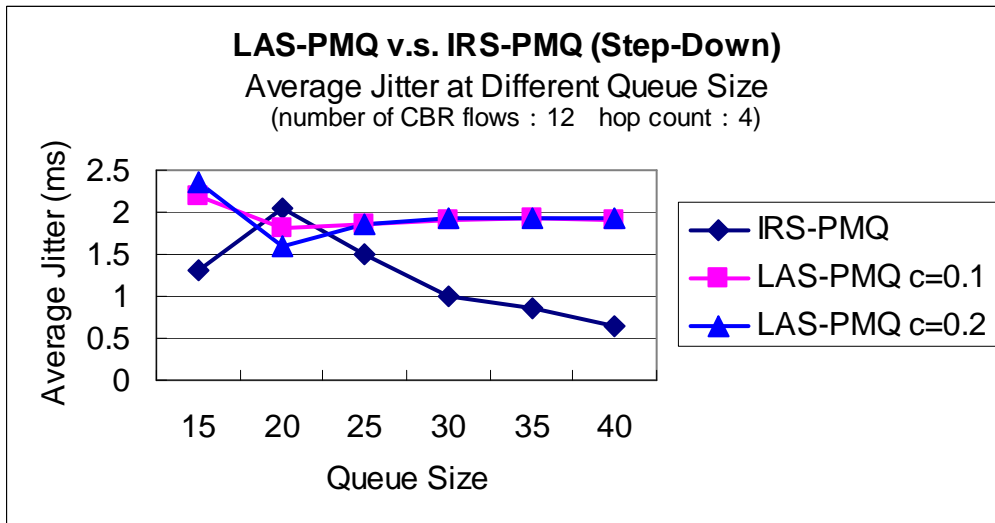
4.5.3.2 實驗流程

與實驗 2A 一樣，圖 4.1 是實驗 2C 的實驗拓樸，在這個實驗中觀察在 queue size 不同的情況下，前瞻式排程演算法的效能表現。

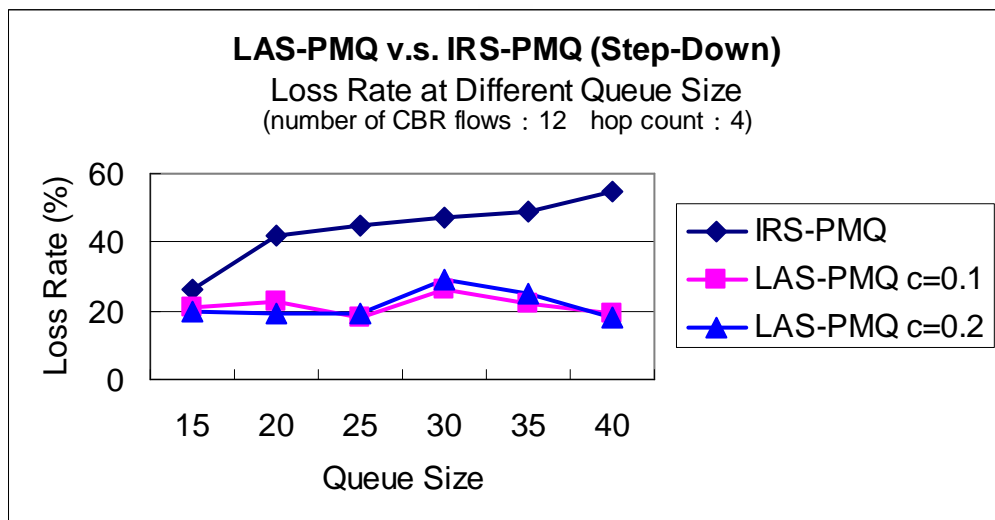
表 4.7 實驗 2C 參數

Parameter	Range
Number of CBR flows	12
Hop count	4
Queue size	15,20,25,30,35,40

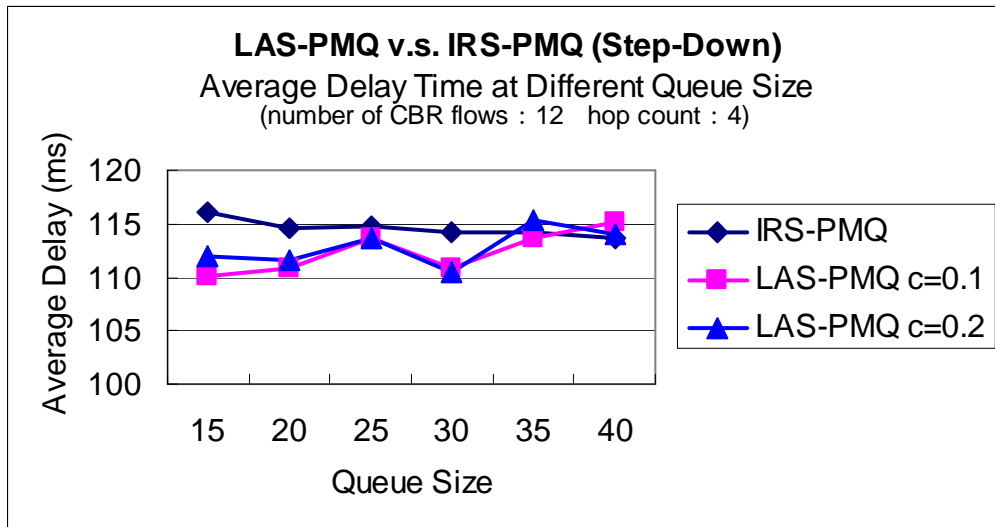
4.5.3.3 實驗結果與分析



(a)



(b)



(c)

圖 4.8 實驗 2C 變動 Queue Size 下 LAS-PMQ 與 IRS-PMQ 之效能比較

(a)Jitter 平均值 (b)封包遺失率 (c)平均延遲時間

圖 4.8 是實驗 2C 的結果。由實驗結果可以看出，在變動 queue size 的情況下，雖然在這個實驗中前瞻式排程的 jitter 在某些情況下不會優於獨立排程，但仍在可接受的範圍。原因是在 queue size 較大的狀況下因為調整 deadline 而改變插入的 queue 影響會較為明顯。而在封包遺失率則有不錯的改善。

綜合實驗 2A、2B 及 2C，整體觀察變動網路負載、路徑長度及 queue size 後，對於前瞻式排程的調整係數，我們選擇 0.1 當作我們前瞻式排程的最佳調整係數，並進行後續實驗。因為調整係數為 0.1 時，在多數變動不同控制變數的實驗下都有較佳的效能表現。

4.5.4 實驗 2D：探討他人負載對前瞻式排程效能的影響

4.5.4.1 實驗目標

為了進一步驗證我們所出的前瞻式排程是能有更好的效能，我們另外設計了魚骨拓樸在

各個節點加入 cross traffic，並調整 queue size 以觀察在有他人負載下的情況下前瞻式排程演算法的效能變化。

4.5.4.2 實驗流程

圖 4.9 是實驗 2D 的實驗拓樸，在這個實驗中，我們讓即時性訊務流傳送端在不同的地方開始傳送做為他人負載，以觀察前瞻式排程在此種拓樸下的效能表現。由於這個實驗拓樸固定，我們僅以 queue size 作為控制變數。

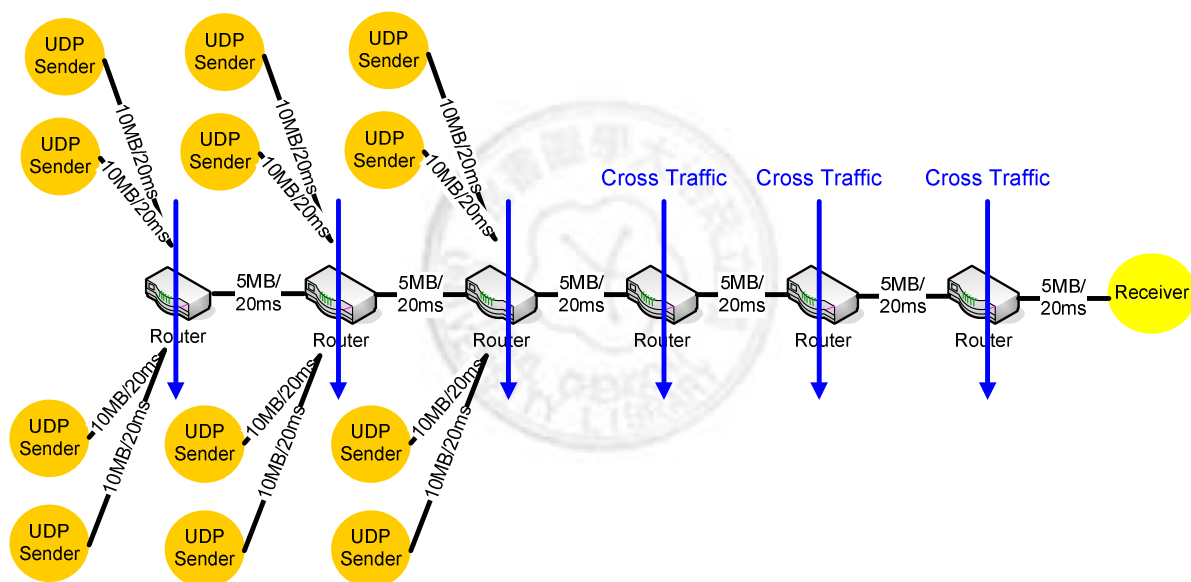
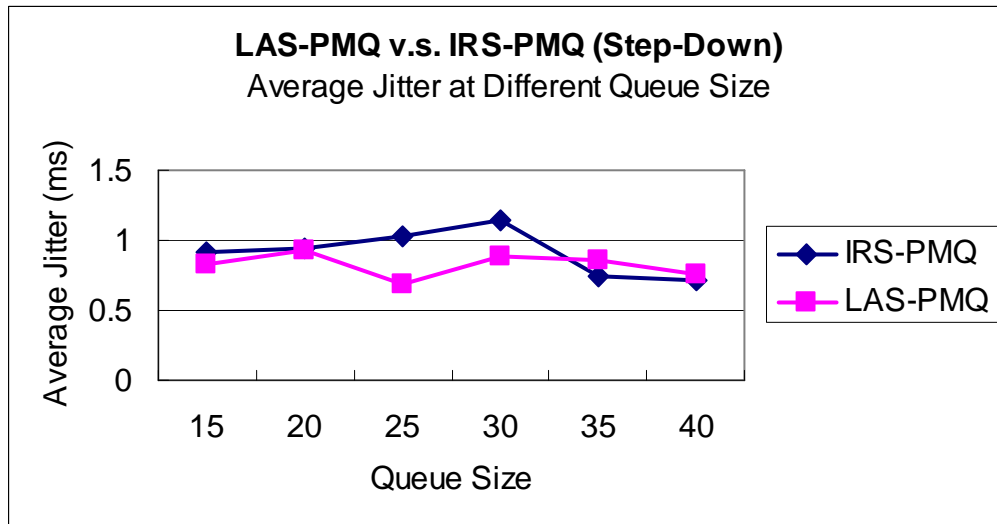


圖 4.9 實驗拓樸二

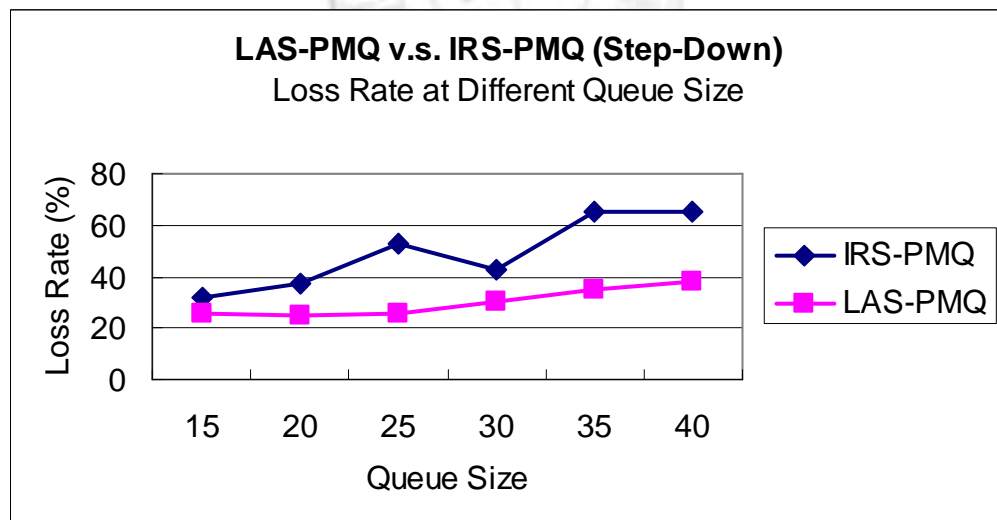
表 4.8 實驗 2D 參數

Parameter	Range
Number of CBR flows	12
Hop count	specified by individual experiments
Queue size	15,20,25,30,35,40

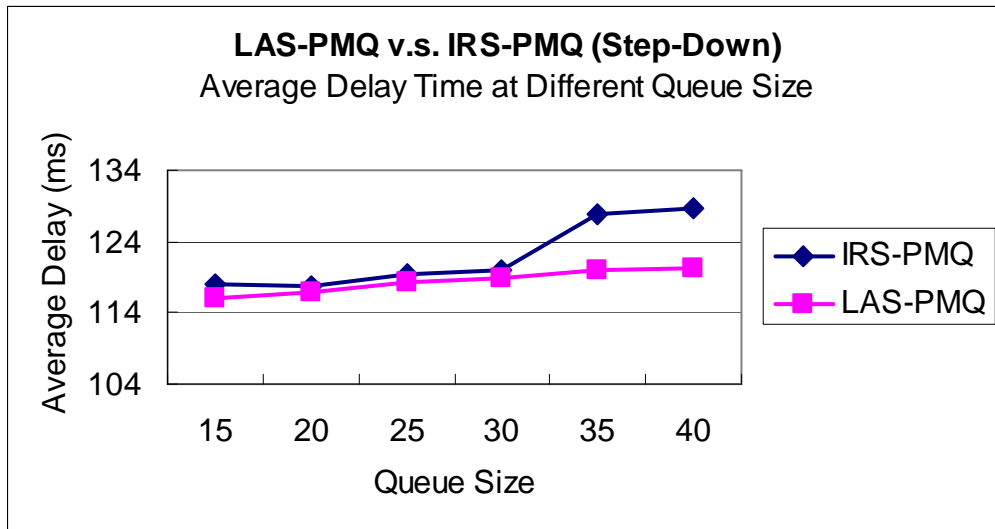
4.5.4.3 實驗結果與分析



(a)



(b)



(c)

圖 4.10 實驗 2D 有他人負載下 LAS-PMQ 與 IRS-PMQ 在變動 Queue Size 時的效能比較

(a)Jitter 平均值 (b)封包遺失率 (c)平均延遲時間

圖 4.10 是實驗 2D 的結果。由實驗結果可以看出前瞻式排程在封包遺失率及平均延遲時間都能有不錯的改善。因為在魚骨拓樸中有的傳送端較為接近接收端，如此這些封包較不緊急可以稍緩送出以補救其他較為緊急的封包。

4.6 實驗三：多服務等級效能評估實驗

在多服務等級的實驗中，我們皆使用前瞻式排程並在 PMQ router 架構下進行。

在實驗 3A 中先以 Streaming 服務等級封包分別賦予不同四種 profit function，以觀察在共存的狀況下，各 profit function 互相競爭頻寬資源的積極性，並可用以作為搭配 scheduling policy 的依據。而後在實驗 3B 中，我們在雙 real-time 服務等級的狀況下將四種 profit function 兩兩搭配，並找出適合 Conversational class 及 Streaming class 的最佳組合。

4.6.1 實驗 3A：探討四種 profit function 共存的效能表現

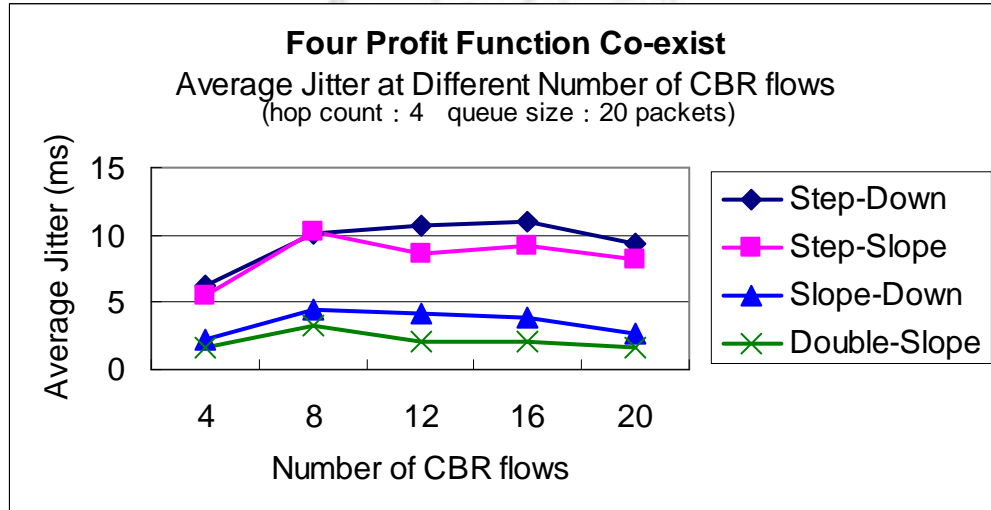
4.6.1.1 實驗目標

調整網路負載並觀察比較四種 profit function 共存時的個別效能變化。

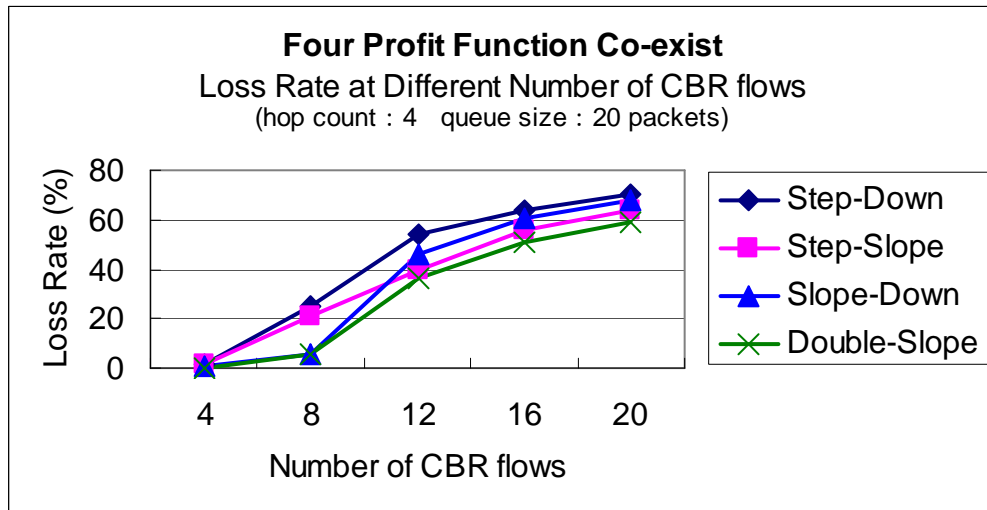
4.6.1.2 實驗流程

圖 4.1 是實驗 3A 的實驗拓樸，在這個實驗中我們逐漸增加即時性訊務個數以觀察在網路負載越來越重的情況下，四種 profit function 的個別效能變化。另外在該拓樸下，路徑長度固定如圖所示，而 queue size 為 20 個封包。

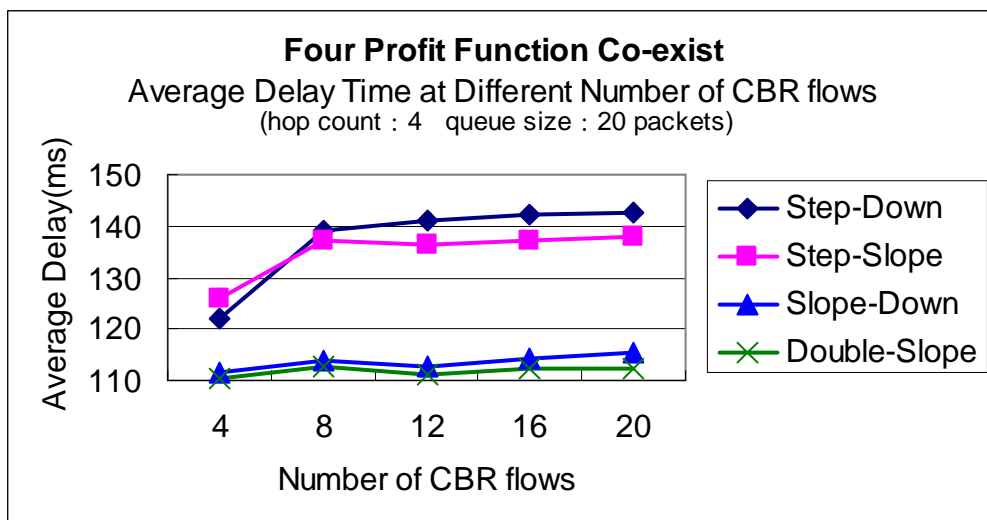
4.6.1.3 實驗結果與分析



(a)



(b)



(c)

圖 4.11 實驗 3A 四種 profit function 共存時變動負載的效能表現
(a)Jitter 平均值 (b)封包遺失率 (c)平均延遲時間

圖 4.11 是實驗 3A 的結果。由實驗結果我們可以看出在 jitter 及 delay time 的部份大致上兩兩分為一組，Step-Down 與 Step-Slope 較差，而 Slope-Down 與 Double-Down 表現較佳，因為 Slope-Down 與 Double-Down 都是在設計上都是越早送出越好，這個結果也可以跟實驗 3B 的結果互相呼應。

4.6.2 實驗 3B：探討雙 real-time 服務等級共存的效能表現

4.6.2.1 實驗目標

在這個實驗中我們逐漸增加 VoIP 的訊務流數量以觀察不同的 VoIP 比例在網路中的整體效能表現，並藉由前述定義的多服務等級評估指標找出使整體網路效能及滿意度最佳的 scheduling policy。

在實驗 3B 中會將我們的方法與 Priority Queue 排程演算法[15]做效能之比較。由[15]中分析 Priority Queue 排程演算法以在實驗中設計逼近 Priority Queue 排程演算法的 scheduler 元件，後文稱為 Simulated Priority Queue (SimPQ)。SimPQ 使用數個 FIFO queue 所組成，每個 queue 服務同一等級的封包，不同等級的封包有不同的修先權，scheduler 必須先服務高優先權才能服務低優先權的封包。

4.6.2.2 實驗流程

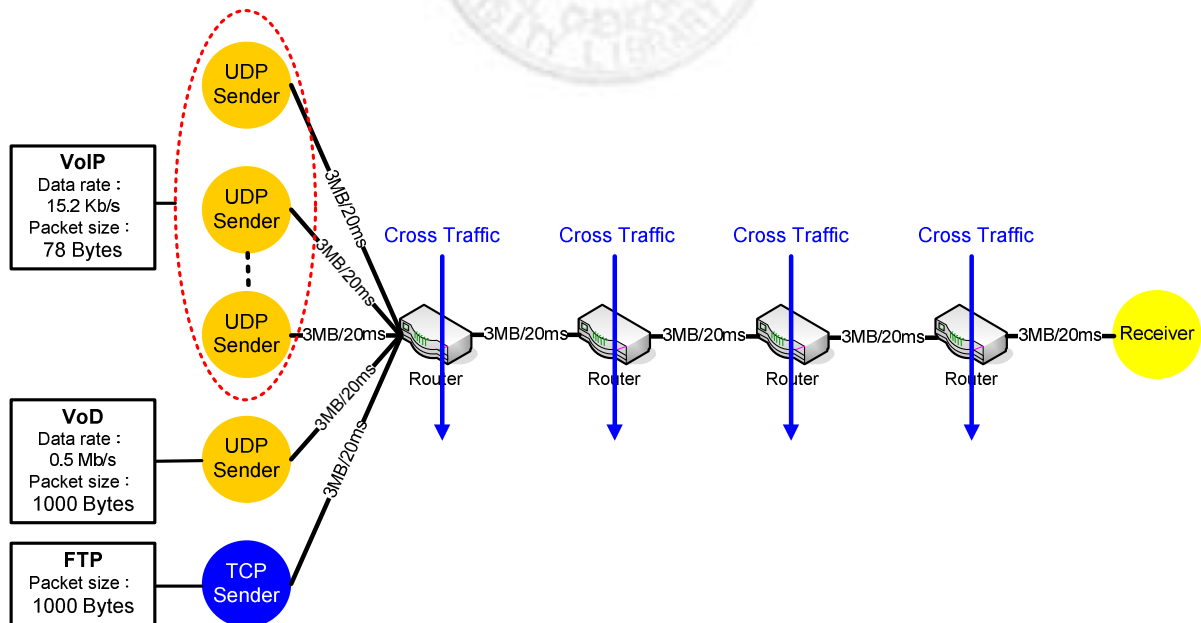


圖 4.12 多服務等級實驗拓樸

圖 4.12 是實驗 3B 的拓樸。在這個實驗中我們模擬雙 real-time 服務等級：VoIP 及 VoD，如圖所示其中 VoIP 訊務流假設以 iLBC 編碼，而其訊務流數量則是這個實驗的控制變數。另外各有一條 VoD 及非 real-time 服務的 FTP 訊務流。另外在該拓樸下，路徑長度固定如圖所示，而 queue size 為 20 個封包。

在雙 real-time 服務等級的情況下，我們將四種 profit function 兩兩組合，之後賦予 Conversational class 及 Streaming class，共有十二種不同的 scheduling policy，如表 4.9 所示。

表 4.9 Scheduling policies for two real-time classes

Scheduling Policy	Conversational Class	Streaming Class
Policy 1	Step-Down	Step-Slope
Policy 2	Step-Down	Slope-Down
Policy 3	Step-Down	Double-Slope
Policy 4	Step-Slope	Slope-Down
Policy 5	Step-Slope	Double-Slope
Policy 6	Slope-Down	Double-Slope
Policy 7	Step-Slope	Step-Down
Policy 8	Slope-Down	Step-Down
Policy 9	Double-Slope	Step-Down
Policy 10	Slope-Down	Step-Slope
Policy 11	Double-Slope	Step-Slope
Policy 12	Double-Slope	Slope-Down

在這個實驗中我們仍然使用前述定義的多服務等級評估指標來衡量不同 scheduling policy 所獲得的 total charge。其中公式(1)與(2)的部份參數設定如下及表 4.10 所示。

$q_3=1$ (quality index for background class)

表 4.10 評估指標各項參數設定

		Conversational	Streaming	Background
Class	i	1	2	3
Quality coefficient	a_i	0.6	0.8	
	b_i	0.2		
	c_i	0.2	0.2	
Charge Threshold	$B_i^{(1)}$	150 ms	1 min	
	$B_i^{(2)}$	15 ms		
	$B_i^{(3)}$	0.05		
Unit Price	C_i	0.001	0.0003	0.0001
Number of flows	N_i	15,20,25,30,35,40	1	1

4.6.2.3 實驗結果與分析

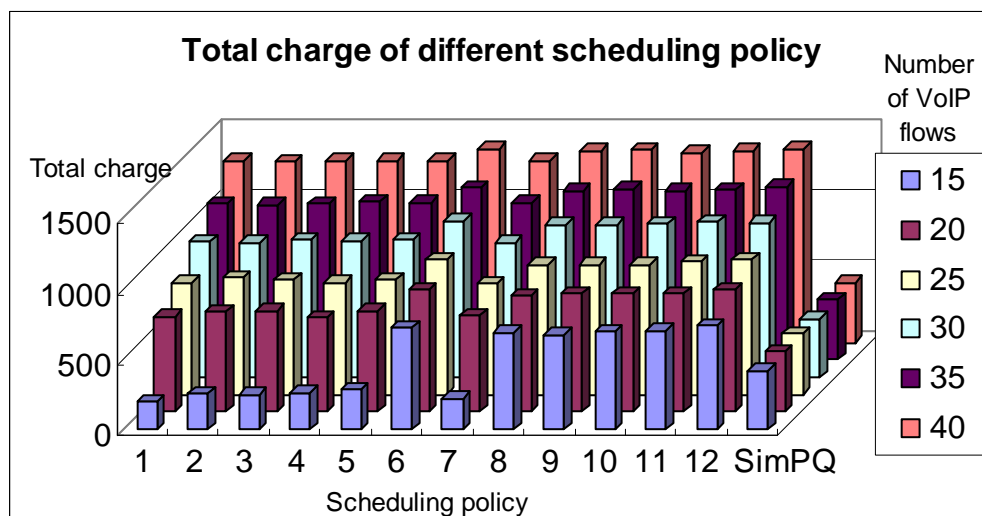
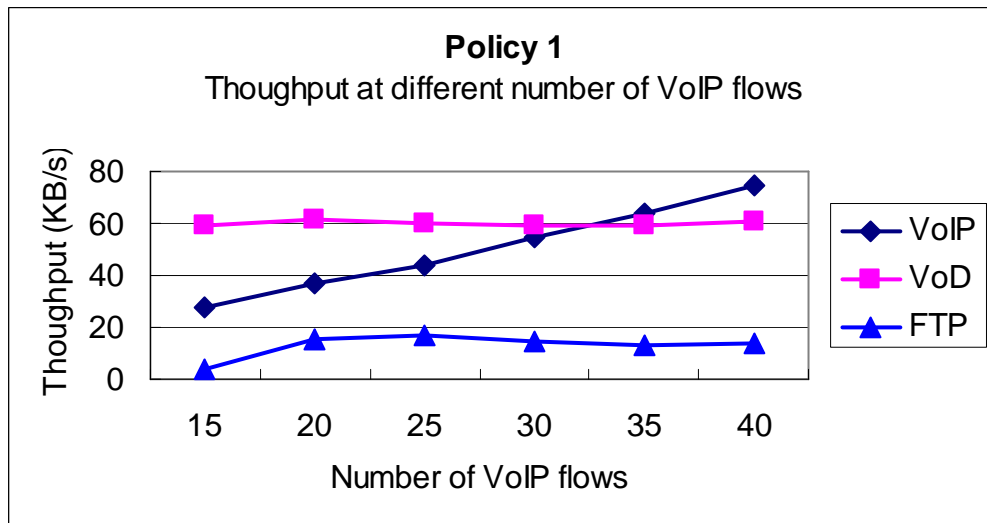


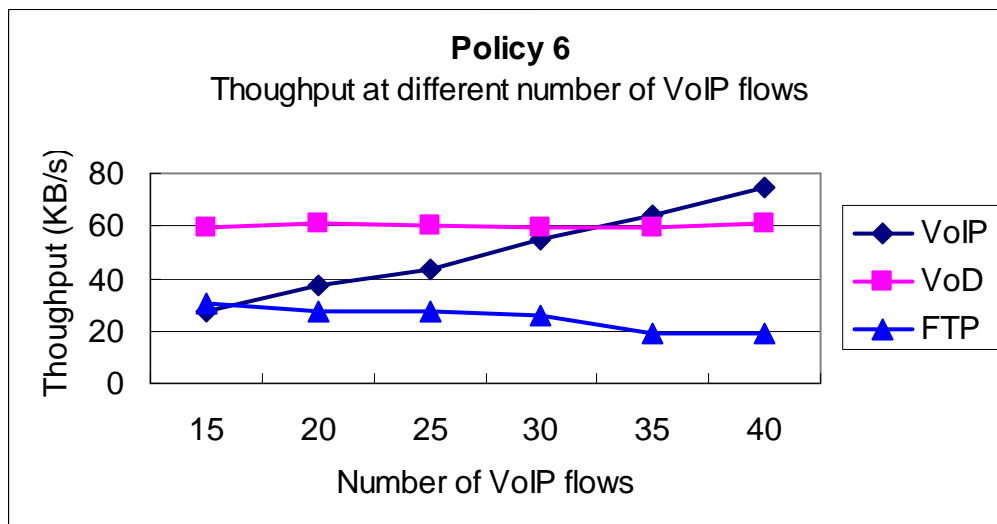
圖 4.13 實驗 3B 雙 real-time 服務等級共存時的效能表現

圖 4.13 是實驗 3B 的結果，其中包含表 4.9 的所有 scheduling policy 的實驗結果，而 SimPQ 是指我們實作的 Simulated Priority Queue 排程演算法。由實驗結果可以看出在我們定義的評估指標下，只要是將 Conversational class 賦予 Slope-Down 或 Double-Slope 的 scheduling policy 整體效能都有不錯的表現。而觀察整體效能的結果，我們推薦 scheduling policy 6，即 Conversational class 的封包賦予 Slope-Down，而 Streaming class 的封包賦予 Double-Slope 的 profit function，這個組合有最佳的整體網路效能。

該實驗結果僅顯示各個 scheduling policy 所獲得的 total charge，因此我們將 throughput、平均封包延遲時間及 jitter 等內部結果分別列出，並將整體效能較差的 scheduling policy 1 一併列出以供比較。



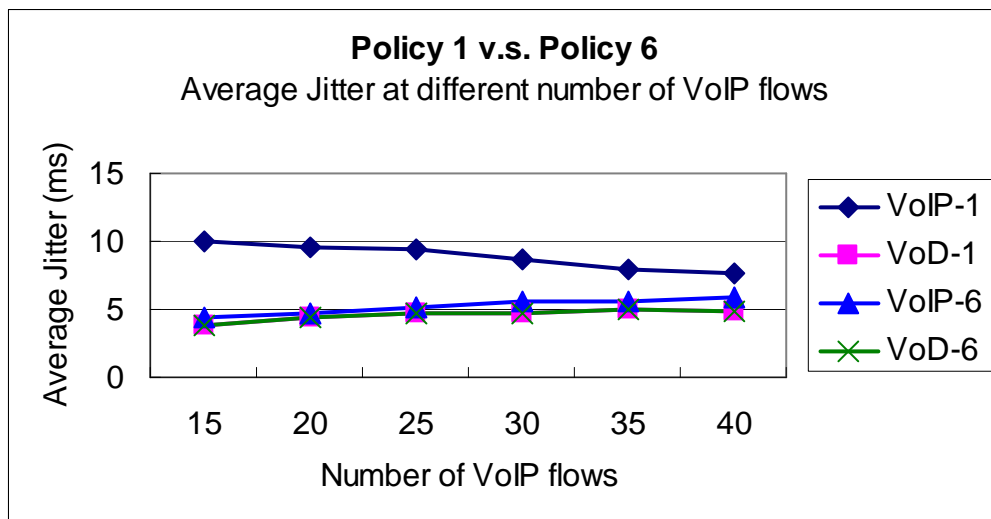
(a)



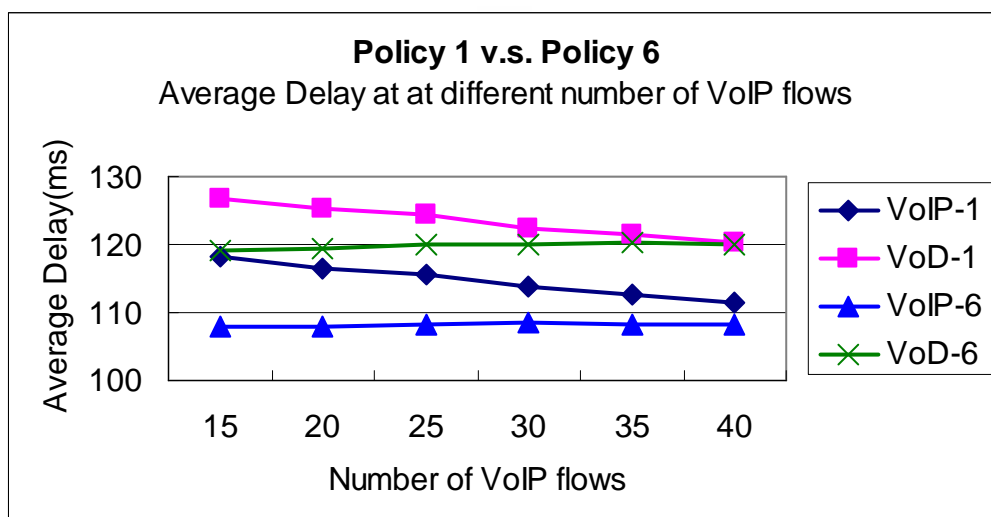
(b)

圖 4.14 實驗 3B 不同等級的 throughput
 (a) Scheduling policy 1 (b) Scheduling policy 6

由圖 4.14 的結果可以看出，當 VoIP 訊務流數量超過 30 條以後，其 throughput 會漸漸超越 VoD，而非 real-time 的 FTP 訊務流則只能搶到少數頻寬。



(a)



(b)

圖 4.15 實驗 3B Scheduling policy 1 與 6 的內部結果比較

(a) Jitter 平均值 (b)平均延遲時間

由圖 4.15 顯示的內部結果我們可以觀察出 scheduling policy 1 整體效能較差的原因，其中賦予 VoIP 服務的 profit function 是 Step-Down，在送出時間可早可晚的特性下，可以看出其 jitter 表現較差，而平均延遲時間更為明顯。Scheduling policy 6 則是都有不

錯的效能表現。

此外，觀察 SimPQ 的實驗結果可以發現在 VoIP 訊務流數量較少(15 條)的情狀下會優於我們方法中較差的 scheduling policy；但在 VoIP 訊務流數量越來越多的情況之下，當服務 VoIP 的 queue 不敷使用以後，則會開始捨棄 VoIP 封包，相當不利於整體滿意度，所以所獲得的 total charge 相當低。在我們的實驗環境、所設計的 charge function 及參數設定下，網路高負載(VoIP 訊務流數量多於 20 條)的情況，我們的方法比 SimPQ 排程演算法能提升 57% 以上的 total charge，如圖 4.13 所示。觀察內部 VoD 訊務流，雖然較我們的方法的 VoD 訊務流可以有較好的效能表現，但是對於整體滿意度來說於事無補。

4.7 實驗總結

在實驗一結果顯示，我們所提出的 PMQ router 架構的確可以改善[13]中的 MQ router 架構，雖然效能仍不及 SPQ router 架構，但是保有硬體容易實作的特性，實際上被採用的機會較大。實驗二結果顯示我們提出的前瞻式排程演算法可以再進一步的改善效能，尤其是在封包遺失率的部份，證明我們依據後續 router 的負載動態調整 profit function 中的 local deadline 可以提升網路的使用率，更可以提高整體網路效能。

實驗三分別賦予 Conversational class 及 Streaming class 不同的 profit function，結果顯示在我們定義的評估指標之下，只要是將 Conversational class 賦予 Slope-Down 或 Double-Slope 的 scheduling policy 整體效能都有不錯的表現。其中又以 scheduling policy 6 能有最佳的整體網路效能，在此 scheduling policy 之下，Conversational class 將優先被服務，有著最高的服務品質，而 Streaming class 雖無最高的服務品質，但是此種 scheduling policy 仍可以達到其服務品質要求，因此能使整體的服務滿意度最高。此外我們的方法讓所有服務等級的封包都共同使用同一個 queue system，因此在資源調配有較大的彈性；但 Priority Queue 的機制並沒有此種彈性，較不利於整體資源調配。

第五章

結論與未來研究方向

Real-time traffic 在現今網路中非常普遍，此類型的訊務對於 jitter 以及 delay time 有著不同的要求，但目前網路是以“Best Effort”的方式處理所有封包，並未根據封包的服務等級差異性的處理，無法滿足 real-time traffic 的品質要求。

本研究延續[13]提出 per-hop 封包排程的方法，利用與封包傳遞時間相關的 profit function 搭配排程演算法，首先改善該研究中的 Multiple FIFO Queue router 架構，並且在每個 router 對封包排程的同時，參考封包在後續路程 router 的負載，改變封包送出的順序，並透過實驗模擬驗證我們方法的效果表現，最後實驗結果顯示我們的方法能有效改善 real-time traffic 的整體效能。並進一步研究在多服務等級之下對不同重要性的封包做差異化的處理方法，提高整體滿意度，我們並透過實驗模擬找出合適的 profit function 組合。在我們的實驗環境及評估指標下，網路高負載的情況，我們的方法與 Simulated Priority Queue 排程演算法比較可以提升 57% 的整體滿意度。原因在於 Simulated Priority Queue 中每種等級的 queue 僅服務一種 class，資源無法互相分享，極可能因為某種等級的服務過多影響其他的服務等級；而我們的方法不但不同 class 間可以互相分享資源，能有較好的資源使用率。而即使是同一種 class 也會因為早到晚到而有所差異，我們的方法也可以做適當的處理。

本研究提出的方法可能仍有不足之處，目前在前瞻式排程環境下僅能在 router 使用 Link-State routing protocol 且 routing metric 為 delay time 的情況下進行，未來將擴展我們

的方法到其他 routing metric，甚至其他 routing protocol，如 Distance Vector 等，則我們必須要找出別的方法來估計後續 router 的負載。而在多服務等級的方面，我們或許可以設計更多適用於不同網路環境的 profit function。未來若能將以上幾點納入考量，將能使我們的方法更加完善。



參考文獻

- [1] M. Alam, R. Prasad, and J.R. Farserotu, “Quality of service among IP-based heterogeneous networks,” *IEEE Personal Communications*, vol. 8, no. 6, pp. 18-24, Dec. 2001.
- [2] J.C.R. Bennett, K. Benson, A. Charny, and J.-Y. Le Boudec, “Delay Jitter Bounds and Packet Scale Rate Guarantee for Expedited Forwarding,” *IEEE/ACM Transactions on Networking*, vol. 10, no. 4, pp. 529-540, Aug. 2002.
- [3] J.C.R. Bennett and H. Zhang, “WF²Q : Worst-case Fair Weighted Fair Queueing,” *Proc. of INFOCOM*, vol. 1, pp.120-128, Mar. 1996.
- [4] D. Black, M. Carlson, E. Davies, and Z. Wang, “An Architecture for Differentiated Services,” *IETF RFC 2475*, Dec. 1998.
- [5] M. Claypool and J. Tanner, “The Effects of Jitter on the Perceptual Quality of Video,” *Proc. of ACM international conference on Multimedia*, pp. 115-118, Nov. 1999.
- [6] E. Crawley, Editor, L. Berger, and S. Berson, “A Framework for Integrated Services and RSVP over ATM,” *IETF RFC 2382*, Aug. 1998.
- [7] T. Dag and O. Gokgol, “A Priority Based Packet Scheduler with Deadline Considerations,” *Proc. of CNSR*, May 2006.
- [8] X. Fei, A. Marshall, and J. Luo, “DO-WF²Q: delay-optimized WF²Q packet scheduling,” *IEEE Proc. on Communications*, vol. 152, no. 5, pp. 655-660, Oct. 2005.

- [9] V.K. Garg and O.T.W. Yu, "Integrated QoS support in 3G UMTS networks," *IEEE Wireless Communications and Networking Conference*, vol. 3, pp. 1187-1192, Sep. 2000.
- [10] Jing-Guo Ge, Ming-Chuan Yang, and Hua-Lin Qian, "Implementation of Expedited Forwarding Using Dynamic Hop Counts Based Absolute Priority Scheduling," *Proc. of ICCT*, vol. 1, pp. 324-333, Apr. 2003.
- [11] Yong Lee, Jian-Yu Lou, Jun-Zhou Luo, and Xiao-Jun Shen, "An Efficient Packet Scheduling Algorithm With Deadline Guarantees for Input-Queued Switches," *IEEE/ACM Transactions on Networking*, vol. 15, no. 1 pp. 212-225, Feb. 2007.
- [12] Yao-Nan Lien, Hung-Ching Jang, Tsu-Chieh Tsai, and Hsing Luh, "Budget Based QoS Management Infrastructure for All-IP Networks," *Proc. of ICACT*, vol. 1, pp. 185-190, Feb. 2005.
- [13] Tsung-Ming Lin, "A New Jitter Control Mechanism by Per-Hop Packet Scheduling Approach," Master thesis, National Chengchi University, Dec. 2006.
- [14] P. Lorenz, "Quality of service and new architectures for future telecommunications networks," *Proc. of MILCOM*, vol. 2, pp. 695-698, Oct. 2000.
- [15] S.-W. Moon, K.G. Shin, and J. Rexford, "Scalable Hardware Priority Queue Architectures for High-Speed Packet Switches," *Proc. of RTAS*, pp. 203-212, Jun. 1997.
- [16] R. Pellizzoni and G. Lipari, "Improved schedulability analysis of real-time transactions with earliest deadline scheduling," *Proc. of RTAS*, pp. 66-75,

Mar. 2005.

- [17] I. Stoica and Hui Zhang, "Providing Guaranteed Services Without Per Flow Management," *Proc. of ACM SIGCOMM*, vol. 29, no. 4, pp. 81-94, Oct. 1999.
- [18] Fu-Ming Tsou, Hong-Bin Chiou, and Zsehong Tsai, "Design and Simulation of an Efficient Real-Time Traffic Scheduler with Jitter and Delay Guarantees," *IEEE Transaction on Multimedia*, vol. 2, no. 4, pp. 255-266, Dec. 2000.
- [19] Jing-Gang Wang and B. Ravindran, "BPA: A Fast Packet Scheduling Algorithm for Real-Time Switched Ethernet Networks," *Proc. of ICPP*, pp. 159-166, 2002.
- [20] Chin-Chi Wu, Hsien-Ming Wu, and Woei Lin, "Efficient and Fair Multi-Level Packet Scheduling for Differentiated Services," *Proc. of ISM*, Dec. 2005.
- [21] Chin-Chi Wu, Hsien-Ming Wu, Chiou Moh, and Woei Lin, "A Hierarchical Packet Scheduler for Differentiated Services in High-Speed Networks," *Proc. of ACIS*, pp. 63-68, Jul. 2006.
- [22] Yan Xu, Yi-Lin Chang, and Zeng-Ji Liu, "Calculation and Analysis of Compensation Buffer Size in Multimedia Systems," *IEEE Communications Letters*, vol. 5, no. 8, pp. 355-357, Aug. 2001.
- [23] Lin-Hua Zhong, Jin Xu, and Xian-Lei Wang, "VWQGRR: a Novel Packet Scheduler," *Proc. of ICN*, pp. 36-36, Apr. 2007.
- [24] 3GPP, "Technical Specification Group Services and Systems Aspects: Architecture for an All IP network," 3GPP TR 23.922 version 1.0.0., Oct.

1999.

[25] “The Network Simulator - ns-2,” <http://www.isi.edu/nsnam/ns/>.

